``

DAFTAR PUSTAKA

A. Engineer. 2015. ***Belajar JSN : Estimasi Konsensus Terdistribusi pada WSN*****,** *http://belajar-jsn.blogspot.co.id*, p. 1, 11. Diakses pada tanggal 10 April 2019.

B.N. Oreshkin, M.J. Coates, and M.G. Rabbat. 2010. ***Optimization and Analiysis of Distributed Averaging With Short Node Memory.*** IEEE TRANSACTION ON SIGNAL PROCESSING. Vol.58, No.5.

Cleve Moler. 2004. ***The Creator of MATLAB The Origins of MATLAB***. http://www.mathworks.com/company/newsletters/news_notes/clevescornr/dec04.html. Diakses pada tanggal 10 April 2019.

D. Artanto. 2012. ***Interaksi Arduino dan Labview***, Elek Media Komputindo,. Jakarta.

Diana, Nazir dan Rufiyanto. 2017. ***Harvesting RF Ambient energy dari End Device LoRa (Long Range Access).*** JURNAL INFOTEL Informatika – Telekomunikasi – Elektronika. Vol.9 No.4:389.

Kazem Sohraby, Daniel Minoli and Taieb Znati. 2007. ***Wireless Sensor Networks, Technology, Protocol, and Application***, Wiley & Sons.

L. Xiao dan S. Boyd. 2004. ***Fast Linear Iterations for Distributed Averaging,*** ELSEIVER, vol. Systems & Control Letters, no. 53, pp. 65-78.

Misbah et al. ***Communication System on Wireless Sensor Networks using Rasberry Pi and Arduino for Monitoring Gas of Air Pollution***. in *Internationl Seminar Intelligent Technology on Its Application*, Denpasar, Bali, 44, p. 39.

Priyambodo, Bambang. 2014. ***Sistem Tata Udara (AHU/HVAC).*** https://priambodo1971.wordpress.com/cpob/sarana-penunjang-kritis-industri-farmasi/sistem-tata-udara-ahuhvac/. Diakses pada tanggal 10 April 2019.

Ramdhani, irwan. 2012. ***Aplikasi Driver Relay Uln2803 Sebagai Penggerak Konveyor Pada Otomatisasi Pengelompokan Buku Menggunakan Inisialisasi Barcode***. Laporan Akhir Jurusan Teknik Elektro Program Studi Elektronika. Palembang: Politeknik Negeri Sriwijaya.

``

Rusli, Andi. 2011. ***Standar Operasi dan Standart Pemeliharaan Switvhgear.*** http://wwwpabriksawitcom.blogspot.com/2011/05/switchgear.html. Diakses pada tanggal 10 April 2019.

Sandeep, Patalay. 2012. ***Railway Signalling Using Wireless Sensor Networ****ks* Sr.IT Engineer, CMC Ltd.

Setiawan, Edi. 2017. ***Algoritma Konsensus Rata-Rata Terdistribusi Pada Wireless Sensor Network Berbasis Link Infrared.*** Thesis Program Magister Jurusan Teknik Elektro Fakultas Teknologi Industri. Surabaya: Institut Teknologi Sepuluh Nopember.

Yan, M., Adiptya, E., & Wibawanto, H. 2013. ***Sistem Pengamatan Suhu dan Kelembaban Pada Rumah Berbasis Mikrokontroller ATmega8***. Jurnal Teknik Elektro, 5(1), 15–17.

``

Lampiran

Lampiran 1 : Source Code Node

Node 1

```
#include <SPI.h>
#include <RH_RF95.h>
#include <LiquidCrystal.h>
#include <dht.h>

#define RFM95_CS 10
#define RFM95_RST 9
#define RFM95_INT 2
#define LED 9
#define RF95_FREQ 915.0

const int rs = 8, en = 7, d4 = 6, d5 = 4, d6 = 1, d7 = 0;
LiquidCrystal lcd(rs, en, d4, d5, d6, d7);
// Singleton instance of the radio driver
RH_RF95 rf95(RFM95_CS, RFM95_INT);

dht sensor;

String getTemp;
String getHum;

int average_temp;
int average_hum;
char hum3[10];
uint8_t temp3[10];
String all_data;
char index=0;
int huml[5];
```

``

Lanjutan Node 1

```
int temp1[5];
int hum2[5];
int temp2[5];


void pairing(void)
{
  //lcd.clear();
  lcd.setCursor(0,1);
  lcd.print(" Pairing........");
  delay(200);
  while(1)
  {
  rf95.send("P", 1);
  delay(10);
  rf95.waitPacketSent();
  uint8_t buf[RH_RF95_MAX_MESSAGE_LEN];
  uint8_t len = sizeof(buf);
  delay(10);
  if (rf95.waitAvailableTimeout(1000))
  {
      //lcd.clear();
      //lcd.setCursor(0,0);
      //lcd.print((char*)buf);
    // Should be a reply message for us now
    if ((rf95.recv(buf, &len))&&(buf[0]='R'))
    {
      lcd.setCursor(0,1);
      lcd.print("   Ready.....   ");

      //delay(2000);
      break;;
    }
    else
    {
      lcd.clear();
      lcd.setCursor(0,0);
      lcd.print("failed");
    }
  }
  }
}

void setup()
{
  pinMode(RFM95_RST, OUTPUT);
  digitalWrite(RFM95_RST, HIGH);
  pinMode(LED, OUTPUT);
  digitalWrite(LED, HIGH);
  lcd.begin(16, 2);
```

Lanjutan Node 1

```
  // manual reset
  digitalWrite(RFM95_RST, LOW);
  delay(10);
  digitalWrite(RFM95_RST, HIGH);
  delay(10);
  lcd.setCursor(0, 1);
  while (!rf95.init()) {
    lcd.print("init failed");
    while (1);
  }
  lcd.print("init OK!");

  if (!rf95.setFrequency(RF95_FREQ)) {
    lcd.clear();
    lcd.print("setF failed");
    while (1);
  }
  rf95.setTxPower(23);
  lcd.clear();
  lcd.setCursor(0,0);
  lcd.print("     NODE 1     ");
  //pairing();
  //lcd.clear();

  delay(1000);
}

void print_average()
{

  int total_temp,total_hum;
  total_temp=0;
  total_hum =0;
  for(int k=0; k<5; k++)
  {
    total_temp += temp1[k]+temp2[k];
    total_hum += hum1[k]+hum2[k];
  }
  average_temp = total_temp/10;
  average_hum = total_hum/10;

  lcd.clear();
  lcd.setCursor(0,0);
  lcd.print("NEW DATA");
  digitalWrite(LED, LOW);
  delay(1000);
  lcd.clear();
  lcd.setCursor(0,0);
  lcd.print("Temperature: ");
  lcd.print(average_temp);
```

``

Lanjutan Node 1

```
  lcd.print(average_temp);
  lcd.setCursor(0,1);
  lcd.print("Humidity    : ");
  lcd.print(average_hum);
  delay(5000);
  digitalWrite(LED, HIGH);
}



void minta_sensor(unsigned char node)
{
  sensor.readl1(A0);
  //templ[index_data] = (int)sensor.temperature;
  //huml[index_data] = (int)sensor.humidity;
  all_data = "N"+(String)node+"#";
  int str_len = all_data.length() + 1;

  char data[str_len];
  all_data.toCharArray(data, str_len);
  rf95.send(data, sizeof(data));
  rf95.waitPacketSent();
}

int H[5][6];
int T[5][6];
int num_node;
int index_dt;
void ambil data()
          _
{
  uint8_t buf[RH_RF95_MAX_MESSAGE_LEN];
  uint8_t len = sizeof(buf);
  if (rf95.waitAvailableTimeout(4000))
  {
    if (rf95.recv(buf, &len))
    {
      if((buf[0]=='N')&&(buf[1]==num_node+48))
      {
        int parsing;
        delay(10);
        //lcd.clear();
        for(int i=0;i<len;i++)
        {
          if(buf[i] == 'N') parsing = 0;
          if(buf[i] == 'T') parsing = 1;
          if(buf[i] == 'H') parsing = 2;
          if(buf[i] == '#') parsing = 3;
          if(parsing == 1)
          {
```

Lanjutan Node 1

```
            if(buf[i]=='T')getTemp = "";
            else
            {
              getTemp+=(char)buf[i];
            }
          }

          if(parsing ==2)
          {
            if(buf[i]=='H')getHum = "";
            else
            {
              getHum+=(char)buf[i];
            }
          }
        }
        T[num_node][index_dt] = getTemp.toInt();
        H[num_node][index_dt] = getHum.toInt();

      }
      else
      {
        //lcd.clear();
        //lcd.print("Receive failed");
        //goto kirim;
      }
    }
  }
  else

  {
    //lcd.clear();
    //lcd.print("noreply  ");
    //lcd.print(num_node);
  }
}

void rata(int persen)
{
  int rata_slave_h=0;
  int rata_slave_t=0;
  int rata_master_h=0;
  int rata_master_t=0;
  int rata_all_h=0;
  int rata_all_t=0;

  for(int x=2; x<=4; x++)
  {
    for(int y=1; y<=5; y++)
    {
```

``

Lanjutan Node 1

```
      rata_slave_h+=H[x][y];
      rata_slave_t+=T[x][y];
    }
  }
  for(int z=1; z<=5; z++)
  {
    rata_master_h+=H[1][z];
    rata_master_t+=T[1][z];
  }
  rata_slave_h/=15;
  rata_slave_t/=15;
  rata_master_h/=5;
  rata_master_t/=5;

  rata_master_h=rata_master_h*persen/100;
  rata_slave_h=rata_slave_h*(100-persen)/100;
  rata_master_t=rata_master_t*persen/100;
  rata_slave_t=rata_slave_t*(100-persen)/100;
  rata_all_h=rata_master_h+rata_slave_h;
  rata_all_t=rata_master_t+rata_slave_t;

  all_data = "AT"+(String)rata_all_t+"H"+(String)rata_all_h+"#";
  int str_len = all_data.length() + 1;
  char data[str_len];
  all_data.toCharArray(data, str_len);
  rf95.send(data, sizeof(data));
  rf95.waitPacketSent();

  lcd.clear();

  lcd.setCursor(0,0);

  lcd.print("NEW DATA");
  digitalWrite(LED, LOW);
  delay(1000);
  lcd.clear();
  lcd.setCursor(0,0);
  lcd.print("Temperature: ");
  lcd.print(rata_all_t);
  lcd.setCursor(0,1);
  lcd.print("Humidity   : ");
  lcd.print(rata_all_h);
  delay(2000);
  digitalWrite(LED, HIGH);

}
```

``

Lanjutan Node 1

```
void loop()
{
  for(index_dt=1;index_dt<=5;index_dt++)
  {
    sensor.readl1(A0);
    digitalWrite(LED, HIGH);
    H[1][index_dt] = (int)sensor.humidity;
    T[1][index_dt] = (int)sensor.temperature;

    for(num_node=2;num_node<=4;num_node++)
    {
      minta_sensor(num_node);
      delay(10);
      ambil_data();
    }
    delay(1000);
  }
  rata(50);
}
```

``

Node 2

```
#include <SPI.h>
#include <RH_RF95.h>
#include <dht.h>
#include <LiquidCrystal.h>

#define RFM95_CS 10
#define RFM95_RST 9
#define RFM95_INT 2

#define RF95_FREQ 915.0
const int rs = 8, en = 7, d4 = 6, d5 = 4, d6 = 1, d7 = 0;
LiquidCrystal lcd(rs, en, d4, d5, d6, d7);

// Singleton instance of the radio driver
RH_RF95 rf95(RFM95_CS, RFM95_INT);

#define LED 9
dht sensor;
int hum2[5];
int temp2[5];
int temp1[5];
int hum1[5];
String ind;
String getTemp;
String getHum;
String all_data;
int index;
int average_temp;
int average hum;

void setup()
{
  pinMode(LED, OUTPUT);
  digitalWrite(LED, HIGH);
  pinMode(RFM95_RST, OUTPUT);
  digitalWrite(RFM95_RST, HIGH);

  lcd.begin(16, 2);

  // manual reset
  digitalWrite(RFM95_RST, LOW);
  delay(10);
  digitalWrite(RFM95_RST, HIGH);
  delay(10);
  lcd.setCursor(0,1);
  while (!rf95.init()) {
    lcd.print("init failed");
    while (1);
  }
  lcd.print("init OK!");
```

``

Lanjutan node 2

```
  // Defaults after init are 434.0MHz, modulation GFSK_Rb250Fd250, +13dbM
  if (!rf95.setFrequency(RF95_FREQ)) {
    lcd.print("setF failed");
    while (1);
  }
  rf95.setTxPower(23);
  lcd.clear();
  lcd.setCursor(0,0);
  lcd.print("     NODE 2     ");
  lcd.setCursor(0,1);
  lcd.print("Ready...");
  delay(1000);
}

void send_sensor(char index_data)
{
  sensor.readll(A0);
  digitalWrite(LED, HIGH);
  hum2[index_data] = (int)sensor.humidity;
  temp2[index_data] = (int)sensor.temperature;
  all_data = "N2"+ind+"T"+(String)temp2[index_data]+"H"+(String)hum2[index_data]+"#";
  int str_len = all_data.length() + 1;
  char data[str_len];
  all_data.toCharArray(data, str_len);
  rf95.send(data, sizeof(data));
  rf95.waitPacketSent();
}


void print_average()
{
  int total_temp,total_hum;
  total_temp=0;
  total_hum =0;
  for(int k=0; k<5; k++)
  {
    total_temp += temp1[k]+temp2[k];
    total_hum += hum1[k]+hum2[k];
  }
  average_temp = total_temp/10;
  average_hum = total_hum/10;

  lcd.clear();
  lcd.setCursor(0,0);
  lcd.print("NEW DATA");
  digitalWrite(LED, LOW);
  delay(1000);
  lcd.clear();
  lcd.setCursor(0,0);
  lcd.print("Temperature: ");
```

``

Lanjutan Node 2

```
  lcd.print(average_temp);
  lcd.setCursor(0,1);
  lcd.print("Humidity    : ");
  lcd.print(average_hum);
  delay(5000);
  digitalWrite(LED, HIGH);
}

void get_data()
{
  uint8_t buf[RH_RF95_MAX_MESSAGE_LEN];
  uint8_t len = sizeof(buf);
  if (rf95.available())
  {
    if (rf95.recv(buf, &len))
    {
      if((buf[0]=='N')&&(buf[1]=='1'))
      {
        ind = (char)buf[2];
        index = ind.toInt();
        send_sensor(index);
        int parsing;
        delay(10);
        for(int i=0;i<len;i++)
        {
          if(buf[i] == 'N') parsing = 0;
          if(buf[i] == 'T') parsing = 1;
          if(buf[i] == 'H') parsing = 2;

          if(parsing == 1)
          {
            if(buf[i]=='T')getTemp = "";
            else
            {
              getTemp+=(char)buf[i];
            }
          }

          if(parsing ==2)
          {
            if(buf[i]=='H')getHum = "";
            else
            {
              getHum+=(char)buf[i];
            }
          }
        }
        templ[index] = getTemp.toInt();
        huml[index] = getHum.toInt();
        if(index==4)
        {
```

Lanjutan Node 2

```
            delay(2000);
            print_average();
          }
        }
        else if ((buf[0]=='P'))
        {
          delay(100);
          rf95.send("Q",1);
          rf95.waitPacketSent();
        }
        else
        {
          //lcd.clear();
          //lcd.print("Receive failed");
        }
      }
      else
      {
        lcd.clear();
        lcd.print("noreply");
      }
    }
}

int H,T;

void kirim_data()
{
  sensor.readl1(A0);

  digitalWrite(LED, HIGH);
  H = (int)sensor.humidity;
  T = (int)sensor.temperature;
  all_data = "N2T"+(String)T+"H"+(String)H+"#";
  int str_len = all_data.length() + 1;
  char data[str_len];
  all_data.toCharArray(data, str_len);
  rf95.send(data, sizeof(data));
  rf95.waitPacketSent();
}

int rata_H,rata_T;

void node2()
{
  if (rf95.available())
  {
    uint8_t buf[RH_RF95_MAX_MESSAGE_LEN];
    uint8_t len = sizeof(buf);
    if (rf95.recv(buf, &len))
    {
```

Lanjutan Node 2

```
if((buf[0]=='N')&&(buf[1]=='2'))
{
  kirim_data();
}
else if(buf[0]=='A')
{
  digitalWrite(LED,LOW);
  int parsing;
  delay(10);
  //lcd.clear();
  for(int i=0;i<len;i++)
  {
    if(buf[i] == 'A') parsing = 0;
    if(buf[i] == 'T') parsing = 1;
    if(buf[i] == 'H') parsing = 2;
    if(buf[i] == '#') parsing = 3;
    if(parsing == 1)
    {
      if(buf[i]=='T')getTemp = "";
      else
      {
        getTemp+=(char)buf[i];
      }
    }

    if(parsing ==2)
    {
      if(buf[i]=='H')getHum = "";

      else
      {
        getHum+=(char)buf[i];
      }
    }
  }
  rata_T = getTemp.toInt();
  rata_H = getHum.toInt();

  lcd.clear();
  lcd.setCursor(0,0);

  lcd.print("NEW DATA");
```

``

Lanjutan Node 2

```
        digitalWrite(LED, LOW);
        delay(1000);
        lcd.clear();
        lcd.setCursor(0,0);
        lcd.print("Temperature: ");
        lcd.print(rata_T);
        lcd.setCursor(0,1);
        lcd.print("Humidity    : ");
        lcd.print(rata_H);
        //delay(2000);

      }

    }
  }
}



void loop()
{
  node2();

  /*
  if (rf95.available())
  {
    get_data();
  }*/
}
```

``

Node 3

```
#include <SPI.h>
#include <RH_RF95.h>
#include <dht.h>
#include <LiquidCrystal.h>

#define RFM95_CS 10
#define RFM95_RST 9
#define RFM95_INT 2

#define RF95_FREQ 915.0
const int rs = 8, en = 3, d4 = 4, d5 = 5, d6 = 6, d7 = 7;
LiquidCrystal lcd(rs, en, d4, d5, d6, d7);

RH_RF95 rf95(RFM95_CS, RFM95_INT);

#define LED 13

dht sensor;
int hum2[5];
int temp2[5];
int temp1[5];
int hum1[5];
String ind;
String getTemp;
String getHum;
String all_data;
int index;
int average_temp;
int average hum;


void setup()
{
  pinMode(LED, OUTPUT);
  digitalWrite(LED, HIGH);
  pinMode(RFM95_RST, OUTPUT);
  digitalWrite(RFM95_RST, HIGH);

  lcd.begin(16, 2);

  // manual reset
  digitalWrite(RFM95_RST, LOW);
  delay(10);
  digitalWrite(RFM95_RST, HIGH);
  delay(10);
  lcd.setCursor(0,1);
  while (!rf95.init()) {
    lcd.print("init failed");
    while (1);
  }
  lcd.print("init OK!");
```

``

Lanjutan Node 3

```
  // Defaults after init are 434.0MHz, modulation GFSK_Rb250Fd250, +13dbM
  if (!rf95.setFrequency(RF95_FREQ)) {
    lcd.print("setF failed");
    while (1);
  }
  rf95.setTxPower(23, false);
  lcd.clear();
  lcd.setCursor(0,0);
  lcd.print("     NODE 3     ");
  lcd.setCursor(0,1);
  lcd.print("Ready...");
  delay(1000);
}

void send_sensor(char index_data)
{
  sensor.readll(A1);
  digitalWrite(LED, HIGH);
  hum2[index_data] = (int)sensor.humidity;
  temp2[index_data] = (int)sensor.temperature;
  all_data = "N3"+ind+"T"+(String)temp2[index_data]+"H"+(String)hum2[index_data]+"#";
  int str_len = all_data.length() + 1;
  char data[str_len];
  all_data.toCharArray(data, str_len);
  rf95.send(data, sizeof(data));
  rf95.waitPacketSent();
}


void print_average()
{
  int total_temp,total_hum;
  total_temp=0;
  total_hum =0;
  for(int k=0; k<5; k++)
  {
    total_temp += temp1[k]+temp2[k];
    total_hum += hum1[k]+hum2[k];
  }
  average_temp = total_temp/10;
  average_hum = total_hum/10;

  lcd.clear();
  lcd.setCursor(0,0);
  lcd.print("NEW DATA");
  digitalWrite(LED, LOW);
  delay(1000);
  lcd.clear();
  lcd.setCursor(0,0);
```

``

Lanjutan Node 3

```
  lcd.print("Temperature: ");
  lcd.print(average_temp);
  lcd.setCursor(0,1);
  lcd.print("Humidity    : ");
  lcd.print(average_hum);
  delay(5000);
  digitalWrite(LED, HIGH);

}

void get_data()
{
  uint8_t buf[RH_RF95_MAX_MESSAGE_LEN];
  uint8_t len = sizeof(buf);
  if (rf95.available())
  {
    if (rf95.recv(buf, &len))
    {
      if((buf[0]=='N')&&(buf[1]=='2'))
      {
        ind = (char)buf[2];
        index = ind.toInt();
        send_sensor(index);
        int parsing;
        //lcd.print((char*)buf);
        delay(10);
        for(int i=0;i<len;i++)
        {

          if(buf[i] == 'N') parsing = 0;
          if(buf[i] == 'T') parsing = 1;
          if(buf[i] == 'H') parsing = 2;
          if(buf[i] == '#') parsing = 3;
          if(parsing == 1)
          {
            if(buf[i]=='T')getTemp = "";
            else
            {
              getTemp+=(char)buf[i];
            }
          }

          if(parsing ==2)
          {
            if(buf[i]=='H')getHum = "";
            else
            {
              getHum+=(char)buf[i];
            }
          }
        }
```

Lanjutan Node 3

```
        templ[index] = getTemp.toInt();
        huml[index] = getHum.toInt();
        //lcd.clear();lcd.print((char*)buf);
        if(index==4)
        {
          delay(2000);
          print_average();
        }
      }
      else if ((buf[0]=='Q'))
      {
        delay(100);
        rf95.send("R",1);
        rf95.waitPacketSent();
      }
      else
      {
        //lcd.clear();
        //lcd.print(buf[1]);
        //lcd.print("Receive failed");
      }
    }
    else
    {
      lcd.clear();
      lcd.print("noreply");
    }
  }
}

int H,T;

void kirim_data()
{
  sensor.readll(A1);
  digitalWrite(LED, HIGH);
  H = (int)sensor.humidity;
  T = (int)sensor.temperature;
  all_data = "N3T"+(String)T+"H"+(String)H+"#";
  int str_len = all_data.length() + 1;
  char data[str_len];
  all_data.toCharArray(data, str_len);
  rf95.send(data, sizeof(data));
  rf95.waitPacketSent();
}

int rata_H,rata_T;
void node3()
{
  if (rf95.available())
  {
```

Lanjutan Node 3

```
uint8_t buf[RH_RF95_MAX_MESSAGE_LEN];
uint8_t len = sizeof(buf);
if (rf95.recv(buf, &len))
{

  if((buf[0]=='N')&&(buf[1]=='3'))
  {
    kirim_data();
  }
  else if(buf[0]=='A')
  {
    digitalWrite(LED,LOW);
    int parsing;
    delay(10);
    //lcd.clear();
    for(int i=0;i<len;i++)
    {
      if(buf[i] == 'A') parsing = 0;
      if(buf[i] == 'T') parsing = 1;
      if(buf[i] == 'H') parsing = 2;
      if(buf[i] == '#') parsing = 3;
      if(parsing == 1)
      {
        if(buf[i]=='T')getTemp = "";
        else
        {
          getTemp+=(char)buf[i];
        }

      }

      if(parsing ==2)
      {
        if(buf[i]=='H')getHum = "";
        else
        {
          getHum+=(char)buf[i];
        }
      }
    }
    rata_T = getTemp.toInt();
    rata_H = getHum.toInt();

    lcd.clear();
    lcd.setCursor(0,0);
```

Lanjutan Node 3

```
          lcd.print("NEW DATA");
          digitalWrite(LED, LOW);
          delay(1000);
          lcd.clear();
          lcd.setCursor(0,0);
          lcd.print("Temperature: ");
          lcd.print(rata_T);
          lcd.setCursor(0,1);
          lcd.print("Humidity    : ");
          lcd.print(rata_H);
          //delay(2000);


        }

      }
    }
}

void loop()
{

  node3();

  //lcd.print("hoooo");
}
```

``

Node 4

```
#include <SPI.h>
#include <RH_RF95.h>
#include <dht.h>
#include <LiquidCrystal.h>

#define RFM95_CS 10
#define RFM95_RST 9
#define RFM95_INT 2

#define RF95_FREQ 915.0
const int rs = 8, en = 3, d4 = 4, d5 = 5, d6 = 6, d7 = 7;
LiquidCrystal lcd(rs, en, d4, d5, d6, d7);

RH_RF95 rf95(RFM95_CS, RFM95_INT);

#define LED 13

dht sensor;
int hum2[5];
int temp2[5];
int temp1[5];
int hum1[5];
String ind;
String getTemp;
String getHum;
String all_data;
int index;
int average_temp;
int average_hum;

void setup()
{
  pinMode(LED, OUTPUT);
  digitalWrite(LED, HIGH);
  pinMode(RFM95_RST, OUTPUT);
  digitalWrite(RFM95_RST, HIGH);

  lcd.begin(16, 2);

  // manual reset
  digitalWrite(RFM95_RST, LOW);
  delay(10);
  digitalWrite(RFM95_RST, HIGH);
  delay(10);
  lcd.setCursor(0,1);
  while (!rf95.init()) {
    lcd.print("init failed");
    while (1);
  }
```

``

`Lanjutan Node 4

```
  }
  lcd.print("init OK!");

  // Defaults after init are 434.0MHz, modulation GFSK_Rb250Fd250, +13dbM
  if (!rf95.setFrequency(RF95_FREQ)) {
    lcd.print("setF failed");
    while (1);
  }
  rf95.setTxPower(23, false);
  lcd.clear();
  lcd.setCursor(0,0);
  lcd.print("     NODE 4     ");
  lcd.setCursor(0,1);
  lcd.print("Ready...");
  delay(1000);
}

void send_sensor(char index_data)
{
  sensor.readll(Al);
  digitalWrite(LED, HIGH);
  hum2[index_data] = (int)sensor.humidity;
  temp2[index_data] = (int)sensor.temperature;
  all_data = "N3"+ind+"T"+(String)temp2[index_data]+"H"+(String)hum2[index_data]+"#";
  int str_len = all_data.length() + 1;
  char data[str_len];
  all_data.toCharArray(data, str_len);
  rf95.send(data, sizeof(data));
  rf95.waitPacketSent();

}

void print_average()
{
  int total_temp,total_hum;
  total_temp=0;
  total_hum =0;
  for(int k=0; k<5; k++)
  {
    total_temp += temp1[k]+temp2[k];
    total_hum += hum1[k]+hum2[k];
  }
  average_temp = total_temp/10;
  average_hum = total_hum/10;

  lcd.clear();
  lcd.setCursor(0,0);
  lcd.print("NEW DATA");
  digitalWrite(LED, LOW);
  delay(1000);
```

``

Lanjutan Node 4

```
  lcd.clear();
  lcd.setCursor(0,0);
  lcd.print("Temperature: ");
  lcd.print(average_temp);
  lcd.setCursor(0,1);
  lcd.print("Humidity    : ");
  lcd.print(average_hum);
  delay(5000);
  digitalWrite(LED, HIGH);

}

void get_data()
{
  uint8_t buf[RH_RF95_MAX_MESSAGE_LEN];
  uint8_t len = sizeof(buf);
  if (rf95.available())
  {
    if (rf95.recv(buf, &len))
    {
      if((buf[0]=='N')&&(buf[1]=='2'))
      {
        ind = (char)buf[2];
        index = ind.toInt();
        send_sensor(index);
        int parsing;
        //lcd.print((char*)buf);
        delay(10);
        for(int i=0;i<len;i++)

        {
          if(buf[i] == 'N') parsing = 0;
          if(buf[i] == 'T') parsing = 1;
          if(buf[i] == 'H') parsing = 2;
          if(buf[i] == '#') parsing = 3;
          if(parsing == 1)
          {
            if(buf[i]=='T')getTemp = "";
            else
            {
              getTemp+=(char)buf[i];
            }
          }

          if(parsing ==2)
          {
            if(buf[i]=='H')getHum = "";
            else
            {
              getHum+=(char)buf[i];
            }
```

``

Lanjutan Node 4

```
          }
        }
        templ[index] = getTemp.toInt();
        huml[index] = getHum.toInt();
        //lcd.clear();lcd.print((char*)buf);
        if(index==4)
        {
          delay(2000);
          print_average();
        }
      }
      else if ((buf[0]=='Q'))
      {
        delay(100);
        rf95.send("R",1);
        rf95.waitPacketSent();
      }
      else
      {
        //lcd.clear();
        //lcd.print(buf[1]);
        //lcd.print("Receive failed");
      }
    }
    else
    {
      lcd.clear();
      lcd.print("noreply");
    }

  }
}


int H,T;

void kirim_data()
{
  sensor.readll(Al);
  digitalWrite(LED, HIGH);
  H = (int)sensor.humidity;
  T = (int)sensor.temperature;
  all_data = "N4T"+(String)T+"H"+(String)H+"#";
  int str_len = all_data.length() + 1;
  char data[str_len];
  all_data.toCharArray(data, str_len);
  rf95.send(data, sizeof(data));
  rf95.waitPacketSent();
}
```

``

Lanjutan Node 4

```
int rata_H,rata_T;

void node4()
{
  if (rf95.available())
  {
    uint8_t buf[RH_RF95_MAX_MESSAGE_LEN];
    uint8_t len = sizeof(buf);
    if (rf95.recv(buf, &len))
    {

      if((buf[0]=='N')&&(buf[1]=='4'))
      {
        kirim_data();
      }
      else if(buf[0]=='A')
      {
        digitalWrite(LED,LOW);
        int parsing;
        delay(10);
        //lcd.clear();
        for(int i=0;i<len;i++)
        {
          if(buf[i] == 'A') parsing = 0;
          if(buf[i] == 'T') parsing = 1;
          if(buf[i] == 'H') parsing = 2;
          if(buf[i] == '#') parsing = 3;
          if(parsing == 1)

          {
            if(buf[i]=='T')getTemp = "";
            else
            {
              getTemp+=(char)buf[i];
            }
          }

          if(parsing ==2)
          {
            if(buf[i]=='H')getHum = "";
            else
            {
              getHum+=(char)buf[i];
            }
          }
        }
        rata_T = getTemp.toInt();
        rata_H = getHum.toInt();

        lcd.clear();
        lcd.setCursor(0,0);
```

``

Lanjutan Node 4

```
          lcd.print("NEW DATA");
          digitalWrite(LED, LOW);
          delay(1000);
          lcd.clear();
          lcd.setCursor(0,0);
          lcd.print("Temperature: ");
          lcd.print(rata_T);
          lcd.setCursor(0,1);
          lcd.print("Humidity    : ");
          lcd.print(rata_H);
          //delay(2000);


        }

      }
    }
}

void loop()
{

  node4();

  //lcd.print("hoooo");
}
```

``

RIWAYAT HIDUP

Penulis dilahirkan di Gresik pada 15 April 1993 dengan nama Mohamad Tri Aziz Budi Cahyono Putro, merupakan anak ke-3 dari 3 bersaudara dari pasangan Bapak Sueb dan Ibu Mufidah. Penulis memulai pendidikan formalnya di SD Putra Darul Islam Gresik pada tahun 1999-2005, kemudian meneruskan ke sekolah Menengah Pertama di SMP Negeri 4 Gresik pada tahun 2005-2008, dan melanjutkan pendidikan Sekolah Menengah Kejuruan di SMK PGRI 1 Gresik pada tahun 2009-2011, pada tahun 2014 meneruskan pendidikan S1 di Fakultas Teknik Universitas Muhammadiyah Gresik dengan Program Studi teknik Elektro.

Pengalaman kerja penulis adalah sebagai karyawan di PT.Adhi Karya sebagai Drafter Civil pada Ammonia-Urea II Project pada tahun 2016-2018. Penulis bisa dihubungi melalui email triaziz.moh@gmail.com