

LAMPIRAN

LAMPIRAN 1 : Program Arduino

```
#include <OneWire.h>
#include <DallasTemperature.h>
#include <LiquidCrystal.h>
```

```
#define ONE_WIRE_BUS A2
```

```
#define brs_suhu 3
```

```
#define brs_ph 5
```

```
#define min_suhu 0
```

```
#define max_suhu 45
```

```
#define min_ph 0
```

```
#define max_ph 14
```

```
#define dingin 0
```

```
#define normal_suhu 1
```

```
#define panas 2
```

```
#define sangat_asam 0
```

```
#define asam 1
```

```
#define normal_ph 2
```

```
#define basah 3
```

```
#define sangat_basah 4
```

```
#define kosong 0
```

```
#define sedikit 1
```

```

#define normal_out 2
#define banyak 3
#define sangat_banyak 4

#define faktor_pengurang_suhu 1
#define faktor_pengurang_ph 0.1

#define pin_up 31
#define pin_down A12

#define up_on digitalWrite(pin_up,HIGH)
#define up_off digitalWrite(pin_up,LOW)

#define down_on digitalWrite(pin_down,HIGH)
#define down_off digitalWrite(pin_down,LOW)

#define durasi 1

//+++++Var rule dan bobot
double rule[15];
double bobot_suhu[brs_suhu];
double bobot_ph[brs_ph];

double data_set_ph[200];
double data_set_suhu[50];

//+++++membership function
double mf_sen_suhu[3][3] = { {0, 10, 25 }, //dingin{ 20, 28, 35 },//normal

```

```
{ 25, 35, 45 },//panas  
};
```

```
double mf_sen_ph[5][3]={  
    { 0, 1, 4.5 },//SA  
    { 4, 5, 6.5 },//A  
    { 6, 7, 8 },//N  
    { 7.5, 8.5, 9 },//B  
    { 8.5, 10, 14 }//SB  
};
```

```
double mf_ph_up[]={0,50,100,150,200};  
double mf_ph_down[]={0,50,100,150,200};
```

```
long old_millis;  
long kekentalan;  
int pulse;  
int total_rule;  
bool mode;
```

```
int jml_loop_suhu,jml_loop_ph;  
float ph=0;  
float suhu=0;
```

```
#define SensorPin A0 //pH meter Analog output to Arduino Analog Input 0  
#define Offset 0.00 //deviation compensata  
#define samplingInterval 10  
#define printInterval 100  
#define ArrayLenth 20 //times of collection
```

```

int pHArray[ArrayLenth]; //Store the average value of the sensor feedback
int pHArrayIndex=0;
OneWire oneWire(ONE_WIRE_BUS);
DallasTemperature sensors(&oneWire);

const int rs = 12, en = 11, d4 = 10, d5 = 9, d6 = 8, d7 = 7;
LiquidCrystal lcd(rs, en, d4, d5, d6, d7);

void setup() {
Serial.begin(9600);
Serial3.begin(9600);

sensors.begin();
pinMode(pin_up,OUTPUT);
pinMode(pin_down,OUTPUT);

jml_loop_suhu = (max_suhu - min_suhu) / faktor_pengurang_suhu;
jml_loop_ph = (max_ph - min_ph) / faktor_pengurang_ph;
lcd.begin(16, 2);
int index = 0;

//Serial.print("Suhu: ");

for (int i = jml_loop_suhu; i >= 0; i--){
    data_set_suhu[i] = max_suhu - (index*faktor_pengurang_suhu);
    index++;
    // Serial.print(data_set_suhu[i]);
    // Serial.print(",");
    delay(1);
}

```

```

index = 0;
//Serial.println();
//Serial.print("PH: ");
for (int i = jml_loop_ph; i >= 0; i--){
    data_set_ph[i] = max_ph - (index*faktor_pengurang_ph);
    index++;
    //Serial.print(data_set_ph[i]);
    //Serial.print(",");
    delay(1);
}
Serial.println();
}

long last_on;
int minute,second;

// [54;14.16;41.57]
void loop() {
    get_ph();
    sensors.requestTemperatures();
    suhu = sensors.getTempCByIndex(0);

    //Fuzzy Fikasi
    get_bobot_ph (ph, 5, jml_loop_ph);
    get_bobot_suhu(suhu, 3, jml_loop_suhu);

    //Rulebase
    rule[1] = min_val(bobot_suhu[dingin], bobot_ph[sangat_asam]);
    rule[2] = min_val(bobot_suhu[dingin], bobot_ph[asam]);

```

```

rule[3] = min_val(bobot_suhu[dingin], bobot_ph[normal_ph]);
rule[4] = min_val(bobot_suhu[dingin], bobot_ph[basah]);
rule[5] = min_val(bobot_suhu[dingin], bobot_ph[sangat_basah]);

rule[6] = min_val(bobot_suhu[normal_suhu],
bobot_ph[sangat_asam]);
rule[7] = min_val(bobot_suhu[normal_suhu], bobot_ph[asam]);
rule[8] = min_val(bobot_suhu[normal_suhu],
bobot_ph[sangat_basah]);
rule[9] = min_val(bobot_suhu[normal_suhu], bobot_ph[basah]);
rule[10] = min_val(bobot_suhu[normal_suhu],
bobot_ph[normal_ph]);

rule[11] = min_val(bobot_suhu[panas], bobot_ph[sangat_asam]);
rule[12] = min_val(bobot_suhu[panas], bobot_ph[asam]);
rule[13] = min_val(bobot_suhu[panas], bobot_ph[sangat_basah]);
rule[14] = min_val(bobot_suhu[panas], bobot_ph[basah]);
rule[15] = min_val(bobot_suhu[panas], bobot_ph[normal_ph]);

//defuzzi
double total_rule=0;
for (int xx = 1; xx <= 15; xx++){
    total_rule = rule[xx] + total_rule;
}

//Serial.println(rule[xx]);

//Serial.println("=== Total Rule =====");
//Serial.println(total_rule);

//index of Member ship
double defuzzy_up;

```

```
defuzzy_up = rule[1] * mf_ph_up[sangat_banyak] +
    rule[2] * mf_ph_up[banyak] +
    rule[3] * mf_ph_up[kosong] +
    rule[4] * mf_ph_up[kosong] +
    rule[5] * mf_ph_up[kosong] +
    rule[6] * mf_ph_up[banyak] +
    rule[7] * mf_ph_up[sedikit] +
    rule[8] * mf_ph_up[kosong] +
    rule[9] * mf_ph_up[kosong] +
    rule[10] * mf_ph_up[kosong] +
    rule[11] * mf_ph_up[banyak] +
    rule[12] * mf_ph_up[sedikit] +
    rule[13] * mf_ph_up[kosong] +
    rule[14] * mf_ph_up[kosong] +
    rule[15] * mf_ph_up[kosong];
```

```
double defuzzy_down;
```

```
defuzzy_down = rule[1] * mf_ph_down[kosong] +
    rule[2] * mf_ph_down[kosong] +
    rule[3] * mf_ph_down[kosong] +
    rule[4] * mf_ph_down[sedikit] +
    rule[5] * mf_ph_down[banyak] +
    rule[6] * mf_ph_down[kosong] +
    rule[7] * mf_ph_down[kosong] +
    rule[8] * mf_ph_down[sangat_banyak] +
    rule[9] * mf_ph_down[sedikit] +
    rule[10] * mf_ph_down[kosong] +
    rule[11] * mf_ph_down[kosong] +
    rule[12] * mf_ph_down[kosong] +
    rule[13] * mf_ph_down[banyak] +
    rule[14] * mf_ph_down[sedikit] +
```

```

rule[15] * mf_ph_down[kosong];

//Serial.println(defuzzy);

defuzzy_up = defuzzy_up / total_rule;
defuzzy_down = defuzzy_down / total_rule;

if ((millis()-last_on)>1000){
  second++;
  if (second>59){second=0;minute++;}
  last_on=millis();
}

if (minute>=durasi){
  lcd.clear();
  minute=0;
  if (defuzzy_up<=0 && defuzzy_down<=0){up_off;down_off;}
  else if (defuzzy_up>0 && defuzzy_down<=0){
    lcd.setCursor(0,0);
    lcd.print("PH UP AKTIF:");
    lcd.setCursor(0,1);
    lcd.print("ADD: ");
    lcd.print(defuzzy_up);
    lcd.print(" CC ");
    up_on;delay(defuzzy_up*20);down_off;
  }
  else if (defuzzy_up<=0 && defuzzy_down>0){
    lcd.setCursor(0,0);
    lcd.print("PH DWON AKTIF:");
    lcd.setCursor(0,1);

```



```

lcd.print("ADD: ");
lcd.print(defuzzy_down);
lcd.print(" CC ");

up_off;down_on;delay(defuzzy_down*20);
}
up_off;down_off;
delay(2000);
lcd.begin(16, 2);
}
String protokol = "A"+String(suhu)+"B"+String(ph)+"C";
protokol=protokol+protokol.length()+"D";
Serial3.println(protokol);

lcd.setCursor(0,0);
lcd.print("T:");
lcd.print((int)suhu);
lcd.print(" H:");
lcd.print(ph);
lcd.print(" ");
lcd.print(minute);
lcd.print(":");
lcd.print(second);

lcd.print(" ");

lcd.setCursor(0,1);
lcd.print("UP:");
lcd.print(defuzzy_up);
lcd.print(" DW:");

```

```
lcd.print(defuzzy_down);  
lcd.print(" ");  
  
delay(1);  
}
```

