

PROGRAM ALAT PERNYIRAM TANAMAN OTOMATIS

```
byte LED = 2, S0 = 3, S1 = 4, S2 = 5, S3 = 6, OUT = 7;
char* color[] = {"white", "orange", "yellow", "red",
    "green", "blue", "out of range"};
char point[] = "Point color sensor to ";
char object[] = " object";
char submit[] = "Submit 'ok' if it's done";
int learned_colors[3][6];
#define ma 8
#define mun 9
#include <EEPROM.h>
#define pingPin_t A0
#define pingPin A1
#define lim_dpn A2
#define lim_blk A3
int d_ld;
int d_lb;
int det;

int PercentageRed, PercentageGreen,
    PercentageBlue;
float OutOfRange;
#include <LiquidCrystal.h>
#define ta A8
#define tb A9
#define tc A10
#define td A11
int tunda;
int pulse;
int lama;
const int low_f=2; //pwm max
const int low_m=4; //pwm mid
const int low_n=6; //pwm near

const int med_f=4; //pwm max
const int med_m=6; //pwm mid
const int med_n=8; //pwm near

const int high_f=6; //pwm max
const int high_m=8; //pwm mid
const int high_n=10; //pwm near

const int pwm_h = 90;
const int pwm_m = 75;
const int pwm_l = 60;

int hujan;
int timer;

LiquidCrystal lcd(43, 45, 47, 49, 51, 53);

long duration, inches, cm;

int ulang;

int d_a;
int d_b;
int d_c;
int d_d;
int learned[17];

#include <Wire.h>
#include <TimeLib.h>
#include <DS1307RTC.h>

//jadwal 1
const int j_siram_pagi=7;
const int m_siram_pagi=1;

//jadwal 2
const int j_siram_sore=16;
const int m_siram_sore=48;

int pwm_p;

#define pump 13

void setup() {

    pinMode(ma, OUTPUT);
    pinMode(mun, OUTPUT);

    pinMode(pingPin_t, OUTPUT);
    pinMode(pingPin, INPUT);

    pinMode(pwm_p, OUTPUT);

    pinMode(ta, INPUT);
    pinMode(tb, INPUT);
    pinMode(tc, INPUT);
    pinMode(td, INPUT);

    digitalWrite(ta,HIGH);
    digitalWrite(tb,HIGH);
    digitalWrite(tc,HIGH);
    digitalWrite(td,HIGH);

    pinMode(lim_dpn,INPUT);

    pinMode(lim_blk,INPUT);
    digitalWrite(lim_dpn,HIGH);
    digitalWrite(lim_blk,HIGH);

    lcd.begin(16,2);
    lcd.clear();
    Serial.begin(9600);
    TCS3200_Setup();
    lcd.print("Hello:");
```

```

int i_color;
int ad;
for (i_color = 0; i_color < 6; i_color ++ ) {
lcd.clear();

learned_colors[0][i_color] = EEPROM.read(ad);ad++;
learned_colors[1][i_color] = EEPROM.read(ad);ad++;
learned_colors[2][i_color] = EEPROM.read(ad);ad++;

lcd.setCursor(0,0);
lcd.print(learned_colors[0][i_color]);
lcd.print(" baca");
lcd.print(" - ");
lcd.print(learned_colors[1][i_color]);

lcd.setCursor(0,1);
lcd.print(learned_colors[2][i_color]);
lcd.print(" eep ke=");
lcd.print(ad);
lcd.print(" ");
delay(50);
}

// det=2;
}

void loop() {

if(ulang==0){
for(ulang=0;ulang<10;ulang++){
d_a=digitalRead(ta);
d_b=digitalRead(tb);
d_c=digitalRead(tc);
d_d=digitalRead(td);
delay(50);

Serial.print("jika kalibrasi, press 'A' ");
Serial.print("time = ");
Serial.println(ulang);

if(d_c==0){
lcd.setCursor (0,0);
lcd.write ("MODE TEST RUNNING");
det=2;
}

if(d_a==0)
{
learning_mode();
}
}
}
/*

if (det==0)
{
Serial.println("tunggu rtc aktif");
lcd.setCursor (0,0);
lcd.write ("Tunggu RTC Aktif");
lcd.setCursor (0,1);
lcd.write (" ");
det=2;
}

*/
if(det==5){

d_ld=digitalRead(lim_dpn);
d_lb=digitalRead(lim_blk);

lcd.setCursor(0,0);
lcd.write("BACK TO STATION");
mundur();

if(d_lb==LOW){

henti();

lcd.setCursor(0,0);
lcd.write("LIMIT BELAKANG AKTIV!:"");
lcd.setCursor(0,1);
lcd.write("PROSES SELESAI");
det=0;
delay(500);
henti();
}

//pros jalan... scanning
if(det==2){
d_ld=digitalRead(lim_dpn);
d_lb=digitalRead(lim_blk);

if(d_ld==LOW){
digitalWrite(ma,LOW);
digitalWrite(mun,LOW);
lcd.setCursor(0,0);
lcd.write("LIMIT DEPAN AKTIV!!:"");
delay(500);
det=5;
mundur();
}

if(d_ld==HIGH){

lcd.setCursor(0,0);
lcd.write("Robot Jalan....!!:"");

```

```

jalan();
}
}

// while(1) {
//=====rtc=====
//=====
tmElements_t tm;
if (RTC.read(tm) ) {
  Serial.print("Ok, Time = ");
  print2digits(tm.Hour);
  Serial.write(':');
  print2digits(tm.Minute);
  Serial.write(':');
  print2digits(tm.Second);
  Serial.print(", Date (D/M/Y) = ");
  Serial.print(tm.Day);
  Serial.write('/');
  Serial.print(tm.Month);
  Serial.write('/');
  Serial.print(tm.YearToCalendar(tm.Year));
  Serial.println();

  if(det==0){
    lcd.setCursor(0,0);
    lcd.write("Time=");
    lcd.print(tm.Hour);
    lcd.write(":");
    lcd.print(tm.Minute);
    lcd.write(":");
    lcd.print(tm.Second);
    lcd.write(" ");

    lcd.setCursor(0,1);

    lcd.print(j_siram_pagi);
    lcd.write(":");
    lcd.print(m_siram_pagi);
    lcd.write(":");
    lcd.print("00 ");

    lcd.write(" ");
    lcd.print( j_siram_sore);

    lcd.write(":");
    lcd.print(m_siram_sore);
    lcd.write(":");
    lcd.print("00 ");
  }

  if(tm.Hour==j_siram_sore      &&
    tm.Minute==m_siram_sore){
    Serial.write("Aksi jalan");
    Serial.print("sensor akan diaktifkan:");
    det=2;
  }

  }

  if(tm.Hour==j_siram_pagi      &&
    tm.Minute==m_siram_pagi){
    Serial.write("Aksi jalan");
    Serial.print("sensor akan diaktifkan:");
    det=2;
  }
} else {
  if (RTC.chipPresent() ) {
    Serial.println("The DS1307 is stopped. Please
      run the SetTime");
    Serial.println("example to initialize the time
      and begin running.");
    Serial.println();
  } else {
    Serial.println("DS1307 read error! Please check
      the circuitry.");
    Serial.println();
  }
  delay(9000);
}
delay(1000);
}

void print2digits(int number) {
  if (number >= 0 && number < 10) {
    Serial.write('0');
  }
  Serial.print(number);

  //=====
  =====end rtc=====

  hujan=digitalRead(A4);
  if(hujan==0){
    lcd.setCursor(0,0);
    lcd.write("Hujan.....!!");
    delay(2000);
    det=0;
  }

  if(det==3){
    lcd.setCursor(0,0);
    lcd.write("Scanning Warna");
    delay(2000);
    getColor();
    new_sample();
    classify();
    delay(2000);
    det=4;
  }

  if(det==4)
  {

```

```

siram();
}
}

void jalan()
{
  lcd.setCursor(0,1);
  lcd.write("Jalan.....");

  digitalWrite(pingPin_t, LOW);
  delayMicroseconds(2);
  digitalWrite(pingPin_t, HIGH);
  delayMicroseconds(5);
  digitalWrite(pingPin_t, LOW);
  pinMode(pingPin, INPUT);
  duration = pulseIn(pingPin, HIGH);
  cm = microsecondsToCentimeters(duration);
  Serial.print(cm);
  Serial.print("cm");
  Serial.println();

  lcd.setCursor(11,1);
  lcd.print(cm);
  lcd.write("cm ");

  if(cm>=3 && cm <11){
    lcd.setCursor(0,0);
    lcd.write("Siram Dekat");
    delay(400);

    digitalWrite(ma,LOW);
    digitalWrite(mun,LOW);

    lcd.setCursor(0,1);
    lcd.write("Robot Henti....");
    delay(1000);

    lcd.setCursor(0,1);
    lcd.write("baca warna");
    delay(1000);

    lcd.write("Scanning Warna");
    delay(2000);
    getColor();
    new_sample();
    classify();

    pwm_p=60;

    siram();
  }

  if(cm>=11 && cm <16){
    lcd.setCursor(0,0);
    lcd.write("Siram Sedang");
    delay(400);

```

```

digitalWrite(ma,LOW);
digitalWrite(mun,LOW);

  lcd.setCursor(0,1);
  lcd.write("Robot Henti....");
  delay(1000);

  lcd.setCursor(0,1);
  lcd.write("baca warna");
  delay(1000);
  lcd.write("Scanning Warna");
  delay(2000);
  getColor();
  new_sample();
  classify();

  pwm_p=75;
  siram();
}

if(cm>=16 && cm <=25){
  lcd.setCursor(0,0);
  lcd.write("Siram Jauh");
  delay(400);

  digitalWrite(ma,LOW);
  digitalWrite(mun,LOW);

  lcd.setCursor(0,1);
  lcd.write("Robot Henti....");
  delay(1000);

  lcd.setCursor(0,1);
  lcd.write("baca warna");
  delay(1000);

  lcd.write("Scanning Warna");
  delay(2000);

  getColor();
  new_sample();
  classify();

  pwm_p=90;
  siram();
}

void henti()
{
  digitalWrite(ma,LOW);
  digitalWrite(mun,LOW);

  lcd.setCursor(0,1);
  lcd.write("stop.....");
}

```

```

void mundur()
{
  lcd.setCursor(0,1);
  lcd.write("Mundur.....");

  digitalWrite(ma,LOW);
  digitalWrite(mun,HIGH);
}

long      microsecondsToCentimeters(long
microseconds) {
  // The speed of sound is 340 m/s or 29
  microseconds per centimeter.
  // The ping travels out and back, so to find the
  distance of the
  // object we take half of the distance travelled.
  return microseconds / 29 / 2;
}

//PROSEDUR BACA WARNA DAN JARAK AKAN
  MENGHASILKAN NILAI ISIAN UNT
//VAR PULSE DAN TIMER,,,,, DIISI DG NILAI YG
  DIDEKLARASI
void siram()
{
  lcd.setCursor(0,0);
  lcd.write("prosedur Siram.....");
  lcd.setCursor(0,1);
  lcd.write("pwm=");
  lcd.print(pwm_p);

  lcd.write(" t=");
  lcd.print(lama);

  lcd.write("000 ms ");

  analogWrite(pump,pwm_p); //8
  delay(lama*1000);
  //RESET
  analogWrite(pump,0);
  delay(2000);

  //=====
  digitalWrite(mun,LOW);
  digitalWrite(ma,HIGH);
  delay(1000);
  digitalWrite(mun,LOW);
  digitalWrite(ma,LOW);
  //=====
}

```

```

//FUNGSI PEMBACAAN WARNA
void TCS3200_Setup() {
  pinMode(S0,OUTPUT);
  pinMode(S1,OUTPUT);
  pinMode(S2,OUTPUT);
  pinMode(S3,OUTPUT);
  pinMode(LED,OUTPUT);
  pinMode(OUT,INPUT);
}

void TCS3200_On() {
  digitalWrite(LED,HIGH); // switch LED on
  digitalWrite(S0,HIGH); // output frequency scaling
    (100%)
  digitalWrite(S1,HIGH);
  delay(5);
}

void TCS3200_Off() {
  digitalWrite(LED,LOW); // switch LED off
  digitalWrite(S0,LOW); // power off sensor
  digitalWrite(S1,LOW);
}

void NoFilter() {
  digitalWrite(S2,HIGH); // select no filter
  digitalWrite(S3,LOW);
  delay(5);
}

void RedFilter() {
  digitalWrite(S2,LOW); // select red filter
  digitalWrite(S3,LOW);
  delay(5);
}

void GreenFilter() {
  digitalWrite(S2,HIGH); // select green filter
  digitalWrite(S3,HIGH);
  delay(5);
}

void BlueFilter() {
  digitalWrite(S2,LOW); // select blue filter
  digitalWrite(S3,HIGH);
  delay(5);
}

void input() {
  /*
  String inputBuffer = "";
  while (inputBuffer != "ok") {
    inputBuffer = "";
    while (Serial.available() > 0) {
      char ch = Serial.read();

```

```

    inputBuffer += ch;
    delay (10);
}
}
*/
d_a=digitalRead(ta);

d_b=digitalRead(tb);

d_c=digitalRead(tc);

d_d=digitalRead(td);

while(d_b==HIGH)
{
d_b=digitalRead(tb);
}
delay(50);
}
void getColor() {

float    FrequencyClear,    FrequencyRed,
        FrequencyGreen, FrequencyBlue;
TCS3200_On();
NoFilter();
FrequencyClear                =
    float(pulseIn(OUT,LOW,10000));
RedFilter();
FrequencyRed = float(pulseIn(OUT,LOW,10000));
GreenFilter();
FrequencyGreen                =
    float(pulseIn(OUT,LOW,10000));
BlueFilter();
FrequencyBlue = float(pulseIn(OUT,LOW,10000));
TCS3200_Off();
PercentageRed    =    int((FrequencyClear    /
    FrequencyRed) * 100.0);
PercentageGreen    =    int((FrequencyClear    /
    FrequencyGreen) * 100.0);
PercentageBlue    =    int((FrequencyClear    /
    FrequencyBlue) * 100.0);
OutOfRange = 500.0 / FrequencyClear;
Serial.print(" FrequencyClear = ");
Serial.print(FrequencyRed);
Serial.println(" Hz");
Serial.print(" FrequencyRed = ");
Serial.print(FrequencyRed);
Serial.println(" Hz");
Serial.print(" FrequencyGreen = ");
Serial.print(FrequencyGreen);
Serial.println(" Hz");
Serial.print(" FrequencyBlue = ");
Serial.print(FrequencyBlue);
Serial.println(" Hz");
}
void learning_mode() {
    lcd.clear();
    lcd.write("Kalibrasi");
    lcd.setCursor(0,1);
    int ad;
    for(byte i = 0; i < 6; i ++ ) {
        Serial.print(point);
        if(i == 1) Serial.print("an ");
        else Serial.print("a ");
        Serial.print(color[i]);
        Serial.println(object);
        Serial.println(submit);
        Serial.println("");
        lcd.clear();
        lcd.setCursor(0,1);
        lcd.print(color[i]);
        input();
        getColor();
    /*
        learned_colors[0][i] = PercentageRed;
        learned_colors[1][i] = PercentageGreen;
        learned_colors[2][i] = PercentageBlue;
    */
    EEPROM.write(ad,
        PercentageRed);delay(50);learned_colors[0][i
        ] = EEPROM.read(ad);ad++;
    EEPROM.write(ad,
        PercentageGreen);delay(50);learned_colors[1
        ][i] = EEPROM.read(ad);ad++;
    EEPROM.write(ad,
        PercentageBlue);delay(50);learned_colors[2][
        i] = EEPROM.read(ad);ad++;
    lcd.setCursor(0,0);
    lcd.print(learned_colors[0][i]);
    lcd.print(" baca");
    lcd.print(" - ");
    lcd.print(learned_colors[1][i]);
    lcd.setCursor(0,1);
    lcd.print(learned_colors[2][i]);
    lcd.print(" eep ke=");
    lcd.print(ad);
    delay(500);
    }
}
void new_sample() {
    Serial.print(point);
    Serial.print("a new");
    Serial.println(object);
    Serial.println(submit);
    Serial.println("");
    // input();
    getColor();
}
void classify() {
    int i_color;
    int ClosestColor;
    float MaxDiff;
    float MinDiff = 100.0;

```

```

if(OutOfRange < 1.5) ClosestColor = 6; // object
    out of range
else {
    // find nearest neighbor, k = 1
    for (i_color = 0; i_color < 6; i_color ++) {
        // compute Euclidean distances
        float ED = sqrt(pow((learned_colors[0][i_color]
            - PercentageRed),2.0) +
            pow((learned_colors[1][i_color]
            - PercentageGreen),2.0) +
            pow((learned_colors[2][i_color]
            - PercentageBlue),2.0));
        MaxDiff = ED;
        // find minimum distance
        if (MaxDiff < MinDiff) {
            MinDiff = MaxDiff;
            ClosestColor = i_color;
        }
    }
}
Serial.print("Object is "); // print result
Serial.println(color[ClosestColor]);
Serial.println("");
lcd.clear();
lcd.setCursor(0,1);

```

```

lcd.print("Object="); // print result
lcd.print(color[ClosestColor]);

if(color[ClosestColor]=="green"){
    lcd.setCursor(0,1);
    lcd.write("deteksi hijau");
    lama=4;
    delay(2000);
}

if(color[ClosestColor]=="red"){
    lcd.setCursor(0,1);
    lcd.write("deteksi red");
    lama=3;
    delay(2000);
}

if(color[ClosestColor]=="blue"){
    lcd.setCursor(0,1);
    lcd.write("deteksi blue");
    lama=2;
    delay(2000);
}
}

```