

SURAT PENCATATAN CIPTAAN

Dalam rangka perlindungan ciptaan di bidang ilmu pengetahuan, seni dan sastra berdasarkan Undang-Undang Nomor 28 Tahun 2014 tentang Hak Cipta, dengan ini menerangkan:

Nomor dan tanggal permohonan : EC002022115002, 28 Desember 2022

Pencipta

Nama : **Dr. Eni Sumarni, S.TP., M.Si, Priswanto, S.T., M.Eng dkk**
Alamat : Fakultas Pertanian Universitas Jenderal Soedirman, Purwokerto,
JAWA TENGAH, 53122
Kewarganegaraan : Indonesia

Pemegang Hak Cipta

Nama : **LEMBAGA PENELITIAN DAN PENGABDIAN KEPADA
MASYARAKAT (LPPM). UNSOED**
Alamat : Jalan Dr. Soeparno, Grendeng., Purwokerto, JAWA TENGAH,
53122
Kewarganegaraan : Indonesia
Jenis Ciptaan : **Program Komputer**
Judul Ciptaan : **Desain Sistem Embedded Modern Alat Otomatis Hidroponik
Greenhouse Untuk Melon Premium Dan Sayur**

Tanggal dan tempat diumumkan untuk : 12 Desember 2022, di Purwokerto
pertama kali di wilayah Indonesia atau di luar
wilayah Indonesia

Jangka waktu perlindungan : Berlaku selama 50 (lima puluh) tahun sejak Ciptaan tersebut
pertama kali dilakukan Pengumuman.

Nomor pencatatan : 000430746

adalah benar berdasarkan keterangan yang diberikan oleh Pemohon.

Surat Pencatatan Hak Cipta atau produk Hak terkait ini sesuai dengan Pasal 72 Undang-Undang Nomor 28 Tahun 2014 tentang Hak Cipta.



a.n Menteri Hukum dan Hak Asasi Manusia
Direktur Jenderal Kekayaan Intelektual
u.b.
Direktur Hak Cipta dan Desain Industri

Anggoro Dasananto
NIP.196412081991031002

Disclaimer:

Dalam hal pemohon memberikan keterangan tidak sesuai dengan surat pernyataan, Menteri berwenang untuk mencabut surat pencatatan permohonan.

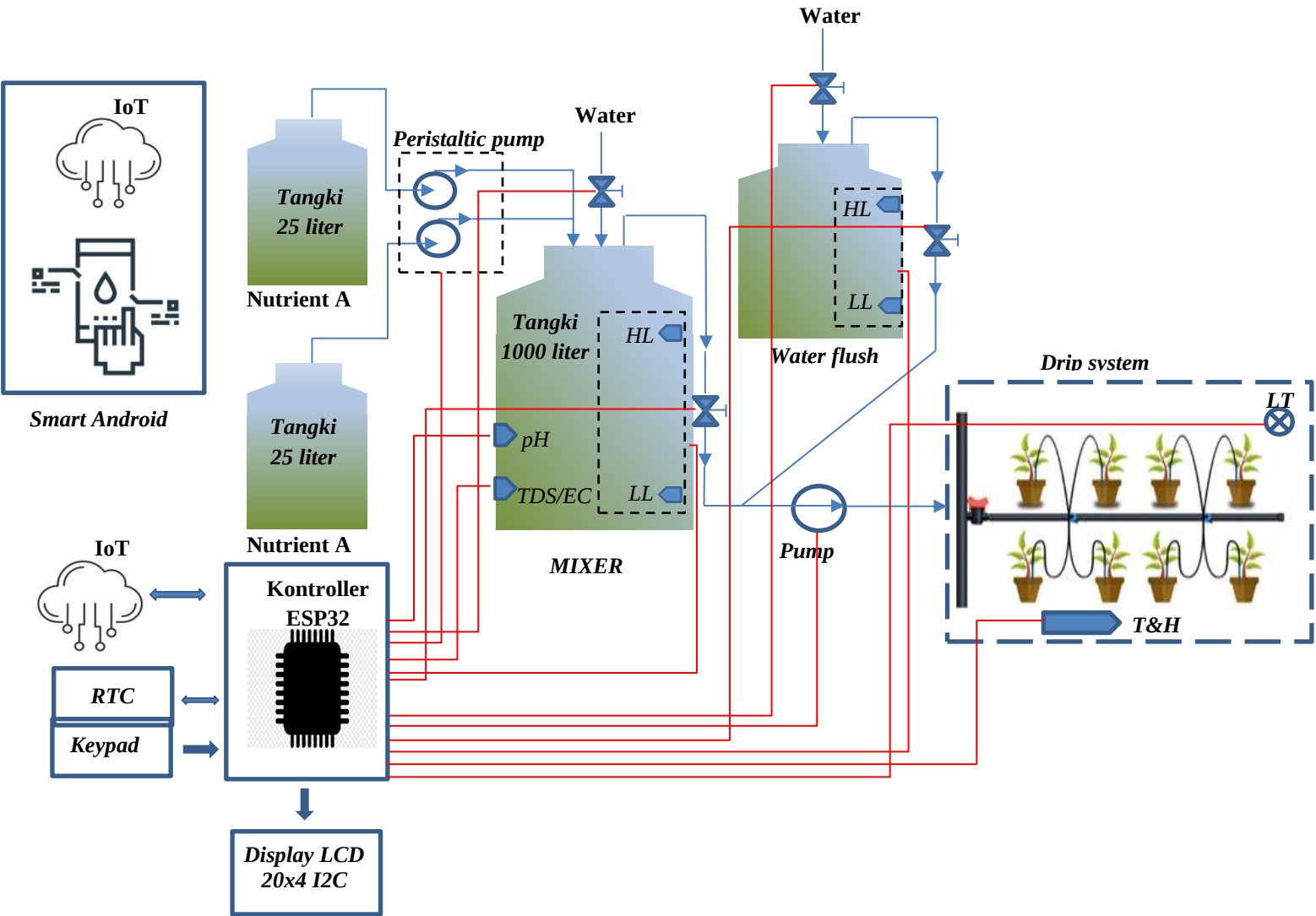
LAMPIRAN PENCIPTA

No	Nama	Alamat
1	Dr. Eni Sumarni, S.TP., M.Si	Fakultas Pertanian Universitas Jenderal Soedirman
2	Priswanto, S.T., M.Eng	Fakultas Teknik Universitas Jenderal Soedirman
3	Widhiatmoko Herry P, S.T., M.T	Fakultas Teknik Universitas Jenderal Soedirman
4	Prof. Ir. Loekas Soesanto, M.S., Ph.D.	Fakultas Pertanian Universitas Jenderal Soedirman
5	Okti Herliana, S.P., M.P.	Fakultas Pertanian Universitas Jenderal Soedirman
6	Ni Wayan Anik Leana, S.P., M.P.	Fakultas Pertanian Universitas Jenderal Soedirman
7	Lutfi Zulkifli, S.P., M.Si	Fakultas Pertanian Universitas Jenderal Soedirman
8	Rizky Sepdi Nugroho	Fakultas Teknik Universitas Jenderal Soedirman
9	Danang Tri Wibowo	Fakultas Teknik Universitas Jenderal Soedirman
10	Febriansyah Dwi Putra	Fakultas Teknik Universitas Jenderal Soedirman



**DESAIN SISTEM EMBEDDED MODERN ALAT OTOMATIS HIDROPONIK
GREENHOUSE UNTUK MELON PREMIUM DAN SAYUR**

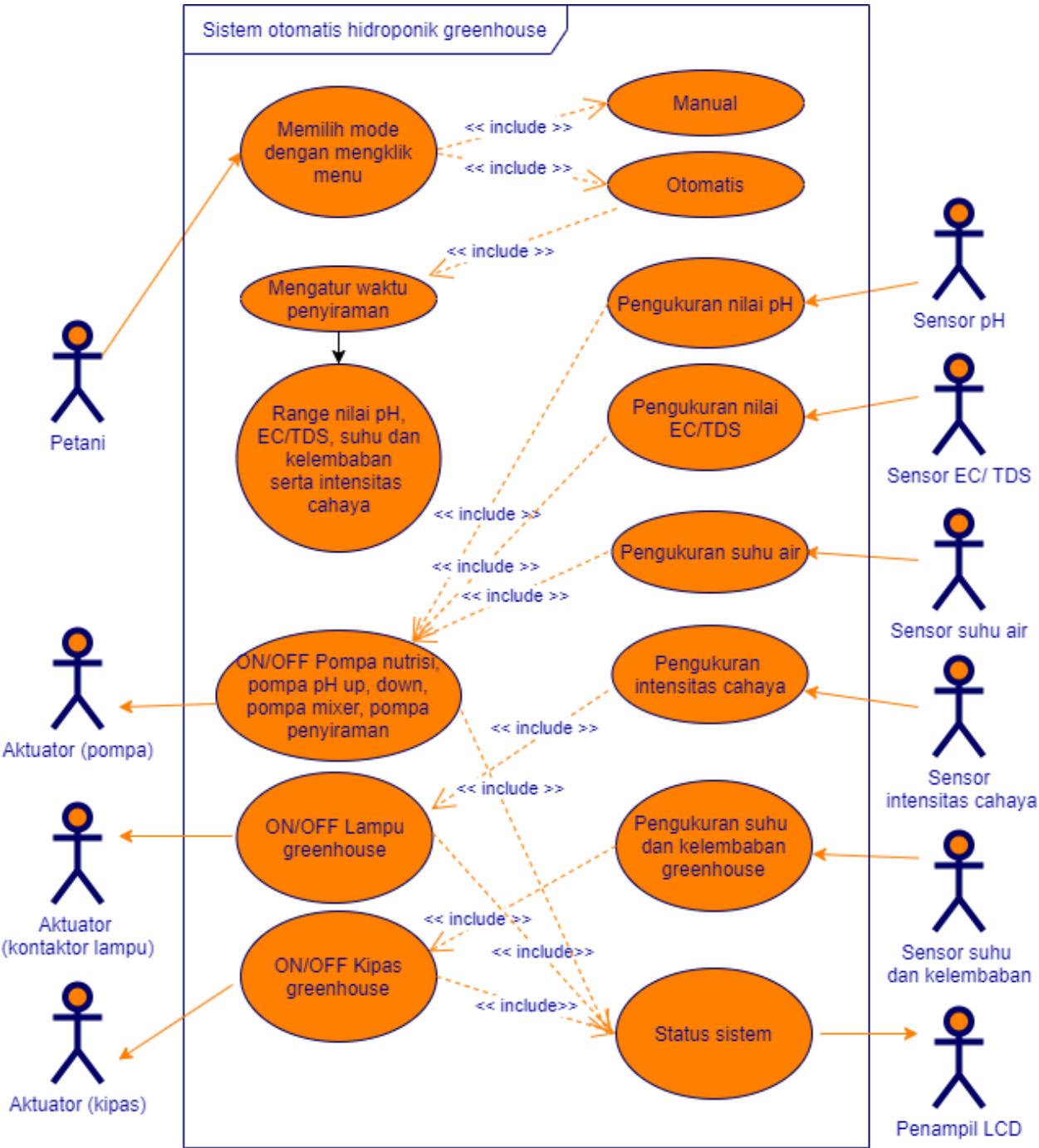
Sistem embedded modern merupakan sistem tertanam yang mengkombinasikan perangkat keras berbasis mikrokontroller dengan perangkat lunak untuk keperluan tertentu. Pada kegiatan Kedaireka Matching Fund ini dilakukan perancangan sistem embedded modern untuk otomatis hidroponik di greenhouse yang diaplikasikan pada tanaman melon premium dan sayur. Teknologi yang di terapkan pada hidroponik ini merupakan sistem kontrol otomatis pemberian nutrisi tanaman dengan dosis yang tepat pada sistem irigasi tetes yang diterapkan pada greenhouse untuk tanaman melon premium dan sayur. Sistem ini terdiri dari : sensor pH, sensor EC/TDS, sensor suhu dan kelembaban, sensor intensitas cahaya sensor level air tangki, serta actuator (pompa). Sistem yang berbasis IoT ini digunakan untuk monitoring dan kontrol real time dengan cara jarak jauh menggunakan aplikasi android maupun website. Sistem secara otomatis akan mengatur komposisi larutan nutrisi sesuai yang di butuhkan pada tanaman (nilai EC= 2.0 – 2.5 mS dan pH = 5.5-6.5). Otomatisasi larutan dilakukan dengan pompa peristaltik yang dapat memberikan takaran dosis nutrisi secara presisi yang akan di campurkan dengan air pada mixer. Kepekatan larutan nutrisi pada mixer di deteksi secara real time menggunakan sensor TDS dan pH. Larutan nutrisi dengan nilai *part per milion* (ppm) yang sesuai akan di alirkan secara otomatis sesuai kebutuhan pada irigasi *drip system*. Pada greenhouse juga di monitor suhu, kelembaban tanaman serta kekeringan tanah. Ketika nutrisi tanaman telah terpenuhi, tetapi kondisi tanah kering, maka *water flush* akan mengalirkan air ke *drip system*. Sistem otomatis ini juga dilengkapi dengan data logger untuk penyimpanan data sensor yang dapat digunakan untuk menganalisa perkembangan tanaman. Diagram blok sistem di tunjukkan pada Gambar 1., berikut .



Gambar 1. Diagram blok sistem otomatis hidroponik

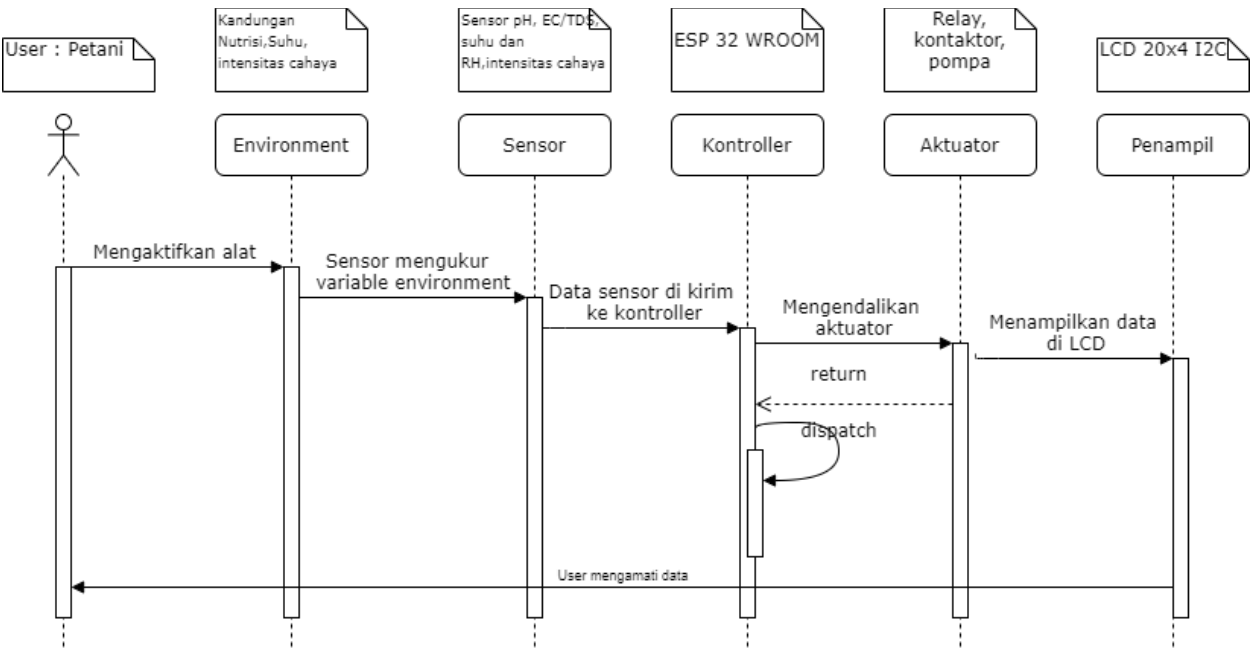
- Keterangan :
- pH = sensor pH, mendeteksi tingkat keasaman air
 - TDS = *Total Dissolve Solid* sensor, untuk mendeteksi tingkat larutan nutrisi
 - T&H = sensor suhu dan kelembaban
 - HL = *high level sensor*
 - LL = *low level sensor*
 - LT = Sensor intensitas cahaya

Desain *use case diagram* Sistem Otomatis Hidroponik Greenhouse :



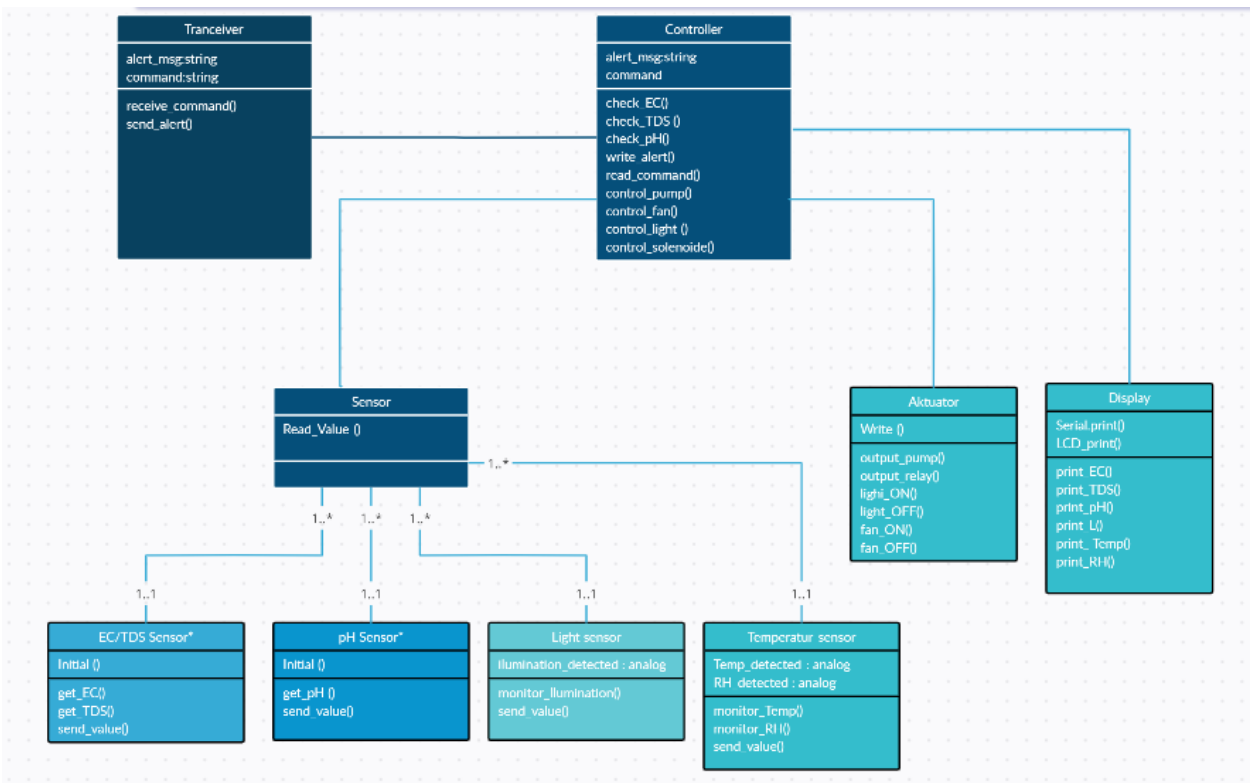
Gambar 2. *Use case diagram* sistem otomatis hidroponik

Desain Sequence Diagram Sistem Otomatis Hidroponik Greenhouse :



Gambar 3. Sequence diagram sistem otomatis hidroponik

Desain Class Diagram Sistem Otomatis Hidroponik Greenhouse :



Gambar 4. Class diagram sistem otomatis hidroponik

Program Sistem Otomatis Hidroponik Greenhouse

```
////////////////////////////////////
// Automation and Monitoring System //
// for Hydroponic Melon Cultivation //
//   in Greenhouse Based on       //
//   Internet of Things           //
////////////////////////////////////
```

```
#include <Arduino.h>
#include <DFRobot_PH.h>
#include <DFRobot_EC.h>
#include <EEPROM.h>
#include <Adafruit_Sensor.h>
#include <WiFi.h>
#include <NTPClient.h>
#include <WiFiUdp.h>
#include <TimeLib.h>
#include <LiquidCrystal_I2C.h>
#include <OneWire.h>
#include <DallasTemperature.h>
```

```
#define EC_PIN 23
#define PH_PIN 15
#define EEPROM_SIZE 256
```

```
#define button1 34
#define button2 35
#define button3 36
#define button4 39
```

```
const char* ssid = "KEDAIREKA";
const char* password = "unsoed1963";
```

```
const int water_temp = 18;
const int pompa_1 = 25;
const int pompa_2 = 26;
const int pompa_3 = 33;
const int pompa_4 = 32;
const int pompa_5 = 17;
const int water_hl = 2;
const int water_ll = 4;
const int drainvalve = 19;
const int relay_cadangan1 = 5;
const int relay_cadangan2 = 16;
```

```
DHT dht(DHTPIN, DHTTYPE);
```

```

OneWire OneWire (water_temp);
DallasTemperature sensors(&OneWire);
DFRobot_EC ec;
DFRobot_PH ph;
LiquidCrystal_I2C lcd(0x27,20,4);
WiFiUDP ntpUDP;
NTPClient timeClient(ntpUDP, "asia.pool.ntp.org", 25200);

char Time[ ] = "00:00:00";
char Date[ ] = "00/00/2000";
byte last_second, second_, minute_, hour_, day_, month_;
int year_;

float ecvoltage,ecValue,phvoltage,phValue, temperature,humidity;
float phmin,phmax,ecmin,ecmax;
float counter,counterph,counterec;

uint8_t hl,ll;

//kondisi button
boolean statusbutton1 = false;
boolean statusbutton2 = false;
boolean statusbutton3 = false;
boolean statusbutton4 = false;

//kondisi akhir button
boolean statusakhirbutton1 = false;
boolean statusakhirbutton2 = false;
boolean statusakhirbutton3 = false;
boolean statusakhirbutton4 = false;

//flag
boolean menu    = false;
boolean kiri    = false;
boolean kanan   = false;
boolean ok      = false;
boolean menyiram = false;
boolean mengaduk = false;

//base
float ph_ref    = 6.0;
float ec_ref[4] = {2.0,2.5};
//int total_ec  = sizeof(ec_ref);
int jamsiram[5] = {8,9,10,11,12};

unsigned long previousMillis    = 0;
const unsigned long interval_sensor = 1000;
unsigned long currentMillis     = 0;

```

```

int halaman_menu = 1;
int halaman_submenu = 1;
int halaman_submenu2 = 1;

void init_wifi(){
  Serial.println("Connecting to ");
  Serial.println(ssid);
  WiFi.begin(ssid, password);

  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  Serial.println("");
  Serial.println("WiFi connected.");
  Serial.println("IP address: ");
  Serial.println(WiFi.localIP());
}

void setup() {
  // put your setup code here, to run once:
  Serial.begin(115200);
  init_wifi();
  ec.begin();
  ph.begin();
  lcd.init();
  lcd.backlight();
  timeClient.begin();
  EEPROM.begin(256);
  sensors.begin();
  pinMode (pompa_1,OUTPUT);
  pinMode (pompa_2,OUTPUT);
  pinMode (pompa_3,OUTPUT);
  pinMode (pompa_4,OUTPUT);
  pinMode (pompa_5,OUTPUT);
  pinMode (water_hl,INPUT);
  pinMode (water_ll,INPUT);
  pinMode (drainvalve,OUTPUT);
  pinMode (relay_cadangan1,OUTPUT);
  pinMode (relay_cadangan2,OUTPUT);
  pinMode (button1,INPUT_PULLUP);
  pinMode (button2,INPUT_PULLUP);
  pinMode (button3,INPUT_PULLUP);
  pinMode (button4,INPUT_PULLUP);
  lcd.clear();
  lcd.setCursor(0,0);
  lcd.print("  GreenHouse  ");
  lcd.setCursor(0,1);
  lcd.print("  Melon Cultivation  ");

```

```

lcd.setCursor(0,2);
lcd.print("Universitas Jenderal");
lcd.setCursor(0,3);
lcd.print("    Soedirman    ");

counterph = EEPROM.get(3,counterph);
phmin     = EEPROM.get(4,phmin);
phmax     = EEPROM.get(5,phmax);
counterec = EEPROM.get(6,counterec);
ecmin     = EEPROM.get(7,ecmin);
ecmax     = EEPROM.get(8,ecmax);

}

void get_clock(){
    timeClient.update();
    unsigned long unix_epoch = timeClient.getEpochTime(); //Get Unix epoch time from the NTP
server
    second_ = second(unix_epoch);
    if (last_second != second_) {

        minute_ = minute(unix_epoch);
        hour_   = hour(unix_epoch);
        day_    = day(unix_epoch);
        month_  = month(unix_epoch);
        year_   = year(unix_epoch);

        Time[7] = second_ % 10 + 48;
        Time[6] = second_ / 10 + 48;
        Time[4] = minute_ % 10 + 48;
        Time[3] = minute_ / 10 + 48;
        Time[1] = hour_   % 10 + 48;
        Time[0] = hour_   / 10 + 48;

        Date[0] = day_   / 10 + 48;
        Date[1] = day_   % 10 + 48;
        Date[3] = month_ / 10 + 48;
        Date[4] = month_ % 10 + 48;
        Date[8] = (year_ / 10) % 10 + 48;
        Date[9] = year_ % 10 % 10 + 48;
    }
}

void button1ditekan(){
    if (statusbutton1 != statusakhirbutton1){
        if (statusbutton1 == 0){
            menu = true;
        }
        delay (50);
    }
    statusakhirbutton1=statusbutton1;
}

```

```
}  
void button2ditekan(){  
    if (statusbutton2 != statusakhirbutton2){  
        if (statusbutton2 == 0){  
            kiri = true;  
        }  
        delay (50);  
    }  
    statusakhirbutton2=statusbutton2;  
}
```

```
void button3ditekan(){  
    if (statusbutton3 != statusakhirbutton3){  
        if (statusbutton3 == 0){  
            //add command  
            kanan = true;  
        }  
        delay (50);  
    }  
    statusakhirbutton3=statusbutton3;  
}
```

```
void button4ditekan(){  
    if (statusbutton4 != statusakhirbutton4){  
        if (statusbutton4 == 0){  
            //add command  
            ok = true;  
        }  
        delay (50);  
    }  
    statusakhirbutton4=statusbutton4;  
}
```

```
void tampil_utama(){  
    if (halaman_menu == 1)  
    {  
        lcd.clear();  
        lcd.setCursor(0,0);  
        lcd.print("Tanggal : ");  
        lcd.setCursor(10,0);  
        lcd.print(Date);  
        lcd.setCursor(0,1);  
        lcd.print("Jam    : ");  
        lcd.setCursor(10,1);  
        lcd.print(Time);  
        lcd.setCursor(0,2);  
        lcd.print("pH : ");  
        lcd.setCursor(5,2);  
        lcd.print(phValue);  
        lcd.setCursor(0,3);
```

```

    lcd.print("EC : ");
    lcd.setCursor(5,3);
    lcd.print(ecValue);
    lcd.setCursor(9,3);
    lcd.print(" mS/cm");
    lcd.setCursor(10,2);
    lcd.print(" C : ");
    lcd.setCursor(15,3);
    lcd.print(temperature);
    last_second = second_;
    delay(500);
}

else if (halaman_menu == 2) //halaman awal
{
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print("Silahkan Memilih :");
    delay(100);
    if (halaman_submenu == 1) //halaman sub_menu ketika button menu ditekan
    {
        lcd.clear();
        lcd.setCursor(0,0);
        lcd.print("Silahkan Memilih :");
        lcd.setCursor(0,1);
        lcd.print("> Atur Nilai pH");
        lcd.setCursor(0,2);
        lcd.print(" Atur Nilai EC");
        lcd.setCursor(0,3);
        lcd.print(" Atur Waktu Mulai");
        delay(100);
    }

    else if (halaman_submenu == 2) //halaman sub_menu ketika button menu ditekan
    {
        lcd.clear();
        lcd.setCursor(0,0);
        lcd.print("Silahkan Memilih :");
        lcd.setCursor(0,1);
        lcd.print(" Atur Nilai pH");
        lcd.setCursor(0,2);
        lcd.print("> Atur Nilai EC");
        lcd.setCursor(0,3);
        lcd.print(" Atur Waktu Mulai");
        delay(100);
    }

    else if (halaman_submenu == 3 ) //halaman sub_menu ketika button menu ditekan
    {
        lcd.clear();

```

```

    lcd.setCursor(0,0);
    lcd.print("Silahkan Memilih :");
    lcd.setCursor(0,1);
    lcd.print("  Atur Nilai pH");
    lcd.setCursor(0,2);
    lcd.print("  Atur Nilai EC");
    lcd.setCursor(0,3);
    lcd.print("> Atur Waktu Mulai");
    delay(100);
}

else if (halaman_submenu = 4 ) //halaman sub_menu ketika button menu ditekan
{
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print("Silahkan Memilih :");
    lcd.setCursor(0,1);
    lcd.print("> Kembali");
    delay(100);
}

}

else if (halaman_menu = 3) //halaman atur ph
{
    lcd.clear();
    lcd.setCursor(1,0);
    lcd.print("Masukan Nilai pH");
    lcd.setCursor(8,1);
    lcd.print("pH : ");
    lcd.setCursor(13,1);
    lcd.print(counterph);
    delay(100);
}

else if (halaman_menu = 4) //halaman atur ec
{
    lcd.clear();
    lcd.setCursor(1,0);
    lcd.print("Masukan Nilai EC");
    lcd.setCursor(8,1);
    lcd.print("EC : ");
    lcd.setCursor(13,1);
    lcd.print(counterec);
    delay(100);
}

else if (halaman_menu = 5) //halaman atur mulai
{
    lcd.clear();

```

```

    lcd.setCursor(0,0);
    lcd.print("Atur Waktu Mulai");
    lcd.setCursor(0,1);
    delay(1000);

}

}

void baca_suhu_air(){
    currentMillis = millis();
    if (millis()- previousMillis >= interval_sensor)
    {

        sensors.requestTemperatures();
        float temperature = sensors.getTempCByIndex(0);
        float temperatureF = sensors.getTempFByIndex(0);
        previousMillis += interval_sensor;
    }

}

void baca_ec(){
    currentMillis = millis();
    if (millis()- previousMillis >= interval_sensor) //time interval: 1s
    {
        ecvoltage = analogRead(EC_PIN)/1024.0*5000; // read the voltage
        ecValue = (1.033*ecvoltage )+ 0.01836;; // convert voltage to EC with temperature
        compensation

        Serial.println("ms/cm");
        previousMillis += interval_sensor;
    }
    ec.calibration(ecvoltage,temperature);
}

void baca_ph(){
    currentMillis = millis();
    if (millis()- previousMillis >= interval_sensor){ //time interval: 1s
        phvoltage = analogRead(PH_PIN)/1024.0*5000; // read the voltage
        phValue = (0.3548*phvoltage) + 3.204; // convert voltage to pH with temperature
        compensation
        Serial.print("^C pH:");
        Serial.println(phValue,2);
        previousMillis += interval_sensor;
    }
    ph.calibration(phvoltage,temperature);
}

void ph_up(){
    digitalWrite(pompa_2,HIGH);

```

```

    currentMillis = millis();
    if (millis()- previousMillis >= interval_sensor*5){
        digitalWrite(pompa_2,LOW);
    }
}

void ph_down(){
    digitalWrite(pompa_3,HIGH);
    currentMillis = millis();
    if (millis()- previousMillis >= interval_sensor*5){
        digitalWrite(pompa_3,LOW);
    }
}

void add_nutrition(){
    digitalWrite(pompa_4,HIGH);
    currentMillis = millis();
    if (millis()- previousMillis >= interval_sensor*15){
        digitalWrite(pompa_5,LOW);
    }
}

void add_water(){
    digitalWrite(pompa_5,HIGH);
    currentMillis = millis();
    if (millis()- previousMillis >= interval_sensor*15){
        digitalWrite(pompa_5,LOW);
    }
}

void drainvalve_on(){
    digitalWrite(drainvalve,HIGH);
    currentMillis = millis();
    if (millis()- previousMillis >= interval_sensor*15){
        digitalWrite(drainvalve,LOW);
        digitalWrite(pompa_1,LOW);
        digitalWrite(pompa_2,LOW);
        digitalWrite(pompa_3,LOW);
        digitalWrite(pompa_4,LOW);
        digitalWrite(pompa_5,LOW);
    }
}

void nyala_siram(){
    int jamsekarang = timeClient.getHours();
    int menitsekarang = timeClient.getMinutes();
    for (int i = 0; i < 5; i++)
    {
        int alarm_set = jamsiram[i];
        if (jamsekarang == alarm_set)

```

```

{
    menyiram = true;
    digitalWrite(relay_cadangan1,HIGH);
    currentMillis = millis();
    if (menitsekarang == menitsekarang+5){
        digitalWrite(relay_cadangan1,LOW);
        menyiram = false;
    }

}

}

}

void utama(){
//cek volume tangki
int jamsekarang = timeClient.getHours();
int menitsekarang = timeClient.getMinutes();

for (int i= 0; i < 5; i++){
    int alarm_cek = jamsiram[i];
    if ((jamsekarang == (alarm_cek - 1))&& (menitsekarang == menitsekarang + 15))
    {
        /* code */
        if (ll == LOW && hl == LOW)
        {
            //isi tangki kosong
            digitalWrite (pompa_1,HIGH);
            if (ll == HIGH && hl == LOW )
            {
                digitalWrite (pompa_1,HIGH);
                if (ll == HIGH && hl == HIGH)
                {
                    digitalWrite(pompa_1,LOW);
                    if (ecValue < ecmin)
                    {
                        add_nutrition();
                    }
                    else if (ecValue > ecmax)
                    {
                        add_water();
                    }
                }
                while ((ecValue >= ecmin)&&(ecValue <= ecmax))
                {
                    if (phValue < phmin)
                    {
                        //jika kadar ph lebih rendah,naikkan ph
                        ph_up();
                    }
                    else if (phValue > phmax)
                    {
                        //jika kadar ph lebih tinggi,turunkan ph

```

```

        ph_down();
    }
    else {
        digitalWrite(drainvalve,LOW);
        digitalWrite(pompa_1,LOW);
        digitalWrite(pompa_2,LOW);
        digitalWrite(pompa_3,LOW);
        digitalWrite(pompa_4,LOW);
        digitalWrite(pompa_5,LOW);
    }
}
}
}
}
else if ((ll == HIGH)&&(hl == HIGH)&&(ecValue > ecmax))
{
    drainvalve_on();
}

else
{
    digitalWrite(drainvalve,LOW);
    digitalWrite(pompa_1,LOW);
    digitalWrite(pompa_2,LOW);
    digitalWrite(pompa_3,LOW);
    digitalWrite(pompa_4,LOW);
    digitalWrite(pompa_5,LOW);
}
}
}
}

void loop() {
    //ambil eeprom
    counterph = EEPROM.get(3,counterph);
    phmin    = EEPROM.get(4,phmin);
    phmax    = EEPROM.get(5,phmax);
    counterec = EEPROM.get(6,counterec);
    ecmin    = EEPROM.get(7,ecmin);
    ecmax    = EEPROM.get(8,ecmax);
    //cekbutton
    statusbutton1 = digitalRead(button1);
    statusbutton2 = digitalRead(button2);
    statusbutton3 = digitalRead(button3);
    statusbutton4 = digitalRead(button4);
    button1ditekan();
    button2ditekan();
    button3ditekan();
    button4ditekan();
}

```

```

//tampil display
tampil_utama();
//kode program
get_clock();
//baca_dht();
baca_ec();
baca_ph();
hl = digitalRead(water_hl);
ll = digitalRead(water_ll);
utama();
nyala_siram();

//tombol menu
if (menu==true)
{
    menu = false;
    halaman_menu = 2;
    halaman_submenu = 1;
}

//tombol kiri
if ((kiri==true) && (halaman_menu == 2))
{
    kiri = false;
    halaman_submenu --;
    if (halaman_submenu < 1)
    {
        halaman_submenu = 4;
    }
}
else if ((kiri == true) && (halaman_menu == 3))
{
    kiri = false;
    counterph -= 0.1;
    if (counterph <= 0.0)
    {
        counterph = 0.0;
    }
}

else if ((kiri == true) && (halaman_menu == 4))
{
    kiri = false;
    counterec -= 0.1;
    if (counterec <= 0.0)
    {
        counterec = 0.0;
    }
}

```

```
//tombol kanan
if ((kanan==true) && (halaman_menu == 2))
{
    kanan = false;
    halaman_submenu++;
    if (halaman_submenu > 4)
    {
        halaman_submenu = 1;
    }
}

else if ((kanan == true) && (halaman_menu == 3))
{
    kanan = false;
    counterph += 0.1;
    if (counterph >= 14.0)
    {
        counterph = 14.0;
    }
}

else if ((kanan == true) && (halaman_menu == 4))
{
    kanan = false;
    counterec += 0.1;
    if (counterec >= 4.0)
    {
        counterec = 4.0;
    }
}

//tombol ok
if (ok==true)
{
    ok = false;
    if (halaman_menu == 2 && halaman_submenu == 1)
    {
        halaman_menu = 3;
    }

    else if (halaman_menu == 2 && halaman_submenu == 2)
    {
        halaman_menu = 4;
    }

    else if (halaman_menu == 2 && halaman_submenu == 3)
    {
        halaman_menu = 5;
    }
}
```

```
else if (halaman_menu == 2 && halaman_submenu == 4)
{
    halaman_menu = 1;
}

else if (halaman_menu = 2)
{
    EEPROM.put(3,counterph);
    phmin = counterph - 0.2;
    EEPROM.put(4,phmin);
    phmax = counterph + 0.2;
    EEPROM.put(5,phmax);
    EEPROM.commit();
    halaman_menu = 1;
}

else if (halaman_menu = 3)
{
    EEPROM.put(6,counterrec);
    ecmin = counterrec - 0.2;
    EEPROM.put(7,ecmin);
    ecmax = counterrec - 0.2;
    EEPROM.put(8,ecmax);
    EEPROM.commit();
    halaman_menu = 1;
}

else if (halaman_menu = 4)
{
    halaman_menu = 1;
}
}
}
```