**LAMPIRAN**

// BACA SUHU DAN KELEMBABAN

float kelembaban = dht.readHumidity();

float suhu = dht.readTemperature();

// BACA SUHU DAN KELEMBABAN

// BACA KETINGGIAN AIR

ketinggianAir = analogRead(analogPin);

ketinggianAirMm = map(ketinggianAir, 0, 1023, 0, 255);

// BACA KETINGGIAN AIR

// BACA WAKTU

DateTime now = rtc.now();

hari = dataHari[now.dayOfTheWeek()];

tanggal = now.day(), DEC;

bulan = now.month(), DEC;

tahun = now.year(), DEC;

jam = now.hour(), DEC;

menit = now.minute(), DEC;

detik = now.second(), DEC;

// BACA WAKTU

// MODUL PAKAN

void modulPakan() {

// proses perputaran pakan secara halus

for (int posisi = 0; posisi <= 180; posisi++) {

// set posisi servo berdasarkan variabel posisi

servoku.write(posisi);

delay(10);

}

// proses perputaran tempat pakan kembali ke posisi awal

for (int posisi = 180; posisi >= 0; posisi--) {

servoku.write(posisi);

delay(10);

}

}

// MODUL PAKAN

// MODUL MINUM

void modulIsiBakMinum() {

// mengisi bak minum dengan menyala pompa air

//relay3

digitalWrite(pompa, relayON);

statusPompa = HIGH;

}

void matikanModulIsiBakMinum() {

// mematikan pompa air

//relay3

digitalWrite(pompa, relayOFF);

statusPompa = LOW;

}

// MODUL MINUM

// MODUL KIRIM DATA

void modulKirimData() {

// BACA SUHU DAN KELEMBABAN

float kelembaban = dht.readHumidity();

float suhu = dht.readTemperature();

// BACA SUHU DAN KELEMBABAN

String dataKirim = "#" + String(ketinggianAirMm) + "#" + String(suhu) + "#" + String(kelembaban) + "#" + String(String() + jam + ":" + menit + ":" + detik) + "#" + String(statusKipasSuhu) + "#" + String(statusKipasKelembaban) + "#" + String(statusPompa) + "#" + String(statusLampu);

Serial.println(dataKirim);

delay(1000);

}

// MODUL KIRIM DATA

void loop(){

// terima data dari hasil komunikasi serial dengan Uno

while(Serial.available() > 0) {

char inChar = (char)Serial.read();

dataIn += inChar;

if (inChar == '\n') {

parsing = true;

}

}

// parsing data yang sudah diterima

if(parsing) {

parsingData();

parsing = false;

dataIn = "";

}

Blynk.run();

timer.run();

}

// MODUL PARSING DATA

void parsingData() {

int j = 0;

Serial.print("data masuk: ");

Serial.print(dataIn);

dt[j] = "";

// perulangan for untuk parsing data

for(i=1; i<dataIn.length(); i++) {

// # sebagai separator antar nilai sensor

// increment j untuk merubah index penampung

if((dataIn[i] == '#')) {

j++;

dt[j] = "";

} else {

dt[j] = dt[j] + dataIn[i];

}

}

}

// MODUL PARSING DATA

// MODUL KIRIM DATA KE BLYNK

void sendSensor(){

// function virtualWrite untuk mengirim data ke pin virtual di Blynk

Blynk.virtualWrite(V0, dt[0].toInt()); // ketinggian air

Blynk.virtualWrite(V1, dt[1].toFloat()); // suhu

Blynk.virtualWrite(V2, dt[2].toFloat()); // kelembaban

Blynk.virtualWrite(V4, dt[4].toInt()); // statusKipasSuhu

Blynk.virtualWrite(V5, dt[5].toInt()); // statusKipasKelembaban

Blynk.virtualWrite(V6, dt[6].toInt()); // statusPompa

Blynk.virtualWrite(V7, dt[7].toInt()); // statusLampu

// function logEvent untuk mengirimkan notifikasi kepada smartphone

if(dt[3] == "7:0:0"){

Blynk.logEvent("status\_pakan","Pakan sudah diberikan pada jam 7");

} else if(dt[3] == "12:0:0"){

Blynk.logEvent("status\_pakan","Pakan sudah diberikan pada jam 12");

} else if(dt[3] == "17:0:0"){

Blynk.logEvent("status\_pakan","Pakan sudah diberikan pada jam 17");

}

}

// MODUL KIRIM DATA KE BLYNK