

LAMPIRAN A

Program Arduino

```
#include <HX711.h>
#include <Wire.h>
#include <RtcDS3231.h>
#include <movingAvg.h>
#include <Servo.h>
#define DOUT 24
#define CLK 26
#define phPin A0
#define turbiPin A1
#define salinitasPin A2
#define voltPin A3
#define arusPin A4
#define motor 2
char daysOfTheWeek[7][12] = {"minggu", "senin",
"selasa", "rabu", "kamis", "jumat", "sabtu"};
int detik, menit, jam, hari;
//-----ph-----
int dataPH;
double Vph;
float phEquals;
float ph_stp;
float vph7 = 3.352
float vph4 = 3.536;
int hasilavg;
//-----turbi-----
int dataTurbi;
int hasilavgTurbi;
double voltTurbi;
double hasilKeruh;
//-----salinitas-----
int dataSalinitas = 0;
int hasilavgSalinitas;
double hasilSalinitas;
//-----loadcell-----
float calibration_factor = 1695.13;
```

```

float gram;
//-----ultrasonik-----
int trigPin = 27;
int echoPin = 29;
long waktu;
int jarak;
int persentase;
//-----sensor arus-----
int sensitivitas = 185; //tegangtung sensor arus yang
digunakan, yang ini 5A
int adcarus= 00;
int teganganoffset = 2500; //nilai pembacaan offset
saat tidak ada arus yang lewat
double tegangan = 00;
double nilaiarus = 00;
//-----sensor Tegangan-----
float Vmodul = 0.0;
float nilaivolt = 0.0;
float R1 = 30000.0; //30k
float R2 = 7500.0; //7500 ohm resistor,
int adcvolt = 0;
//-----global-----
int countGram;
byte countPagi;
int buka = 0;
int tutup = 60;
unsigned long prefTime = 0;
String dataSerial;
String je, me, de, ja, ma, da, js, ms, ds, jm, mm,
dm; //variable jam terima dari serial
int jamPagi, menitPagi, detikPagi, jamSiang,
menitSiang, detikSiang, jamSore, menitSore,
detikSore, jamMalam, menitMalam, detikMalam;
String pakan, arus, volt;
int pakanIn;
RtcDS3231<TwoWire> Rtc(Wire);
HX711 scale;
movingAvg avg(6);

```

```

movingAvg avgTurbi(6);
movingAvg avgSalinitas(6);
Servo openStorage;
Servo openScale;
void setup() {
    Serial.begin(9600);
    Serial1.begin(9600);
    rtcSet();
    pinMode(phPin, INPUT);
    pinMode(turbiPin, INPUT);
    pinMode(salinitasPin, INPUT);
    pinMode(voltPin, INPUT);
    pinMode(arusPin, INPUT);
    pinMode(motor, OUTPUT);
    pinMode(trigPin, OUTPUT);
    pinMode(echoPin, INPUT);
    openStorage.attach(25);
    openScale.attach(23);
    openScale.write(tutup);
    openStorage.write(tutup);
    avg.begin();
    avgTurbi.begin();
    avgSalinitas.begin();
    scale.begin(DOUT, CLK);
    scale.set_scale(calibration_factor);
    scale.tare();
}
void loop() {
    ReadAllSensor();
    printing();
    delay(500);
}
void printing() {
    if ((millis() - prefTime) > 1000) {
        Serial.println("_____");
        Serial.println("countPagi=" + String(countPagi));
        Serial.println("countGrm=" + String(countGram));
        Serial.println("berat=" + String(gram));
    }
}

```

```

        Serial.println(" ");
        Serial.println("adcPH=" + String(hasilavg) + "
Vph=" + String(Vph) + " PH_out=" + String(phEquals));
        Serial.println(" ");
        Serial.println("adcTurbi=" +
String(hasilavgTurbi) + " VTurbi=" +
String(voltTurbi) + " Turbi_out=" +
String(hasilKeruh));
        Serial.println(" ");
Serial1.print("{");
        Serial1.print(String(gram));
        Serial1.print(",");
        Serial1.print(String(phEquals));
        Serial1.print(",");
        Serial1.print(String(hasilKeruh));
        Serial1.print(",");
        Serial1.print(hasilSalinitas, 2);
        Serial1.print(",");
        Serial1.print(String(jam));
        Serial1.print(",");
        Serial1.print(String(menit));
        Serial1.print(",");
        Serial1.print(String(detik));
        Serial1.print(",");
        Serial1.print(String(persentase));
        Serial1.print(",");
        Serial1.print(String(nilaiarus));
        Serial1.print(",");
        Serial1.print(String(nilaivolt));
        Serial1.println("}");
}
        Serial.println("adcSanilitas=" +
String(hasilavgSalinitas ) + " Sal_out=" +
String(hasilSalinitas));
        Serial.println("JAM=" + String(jam) + ":" +
String(menit) + ":" + String(detik));
        Serial.println("Pagi=" + String (jamPagi) + ":" +
String (menitPagi) + ":" + String (detikPagi));

```

```

        Serial.println("siang=" + String (jamSiang) + ":"
+ String (menitSiang) + ":" + String (detikSiang));
        Serial.println("sore=" + String (jamSore) + ":"
String (menitSore) + ":" + String (detikSore));
        Serial.println("Malam=" + String (jamMalam) + ":"
+ String (menitMalam) + ":" + String (detikMalam));
        Serial.println("mam=" + String(pakan));
        Serial.println("Tegangan = " + String(nilaivolt)
+ "ADCVOLT = " + String(adcvolt));
        Serial.println("Arus = " + String(nilaiarus) + "
ADCARUS = " + String(adcarus));
        Serial.println(" ");
        prefTime = millis();
    }
void ReadAllSensor() {
    RtcDateTime now = Rtc.GetDateTime();
    printDateTime(now);
    jam = now.Hour();//jam saat ini
    menit = now.Minute();//menit saat ini
    detik = now.Second();//detik saat ini
    gram = (scale.get_units() * -1), 4;
    //-----rumus ph-----
    dataPH = analogRead(phPin);
    hasilavg = avg.reading(dataPH);
    Vph = 5.0 / 1023.0 * hasilavg;
    ph_stp = (vph4 - vph7) / (7 - 4);
    phEquals = 7.00 + ((vph7 - Vph) / ph_stp);
    //----- rumus turbi-----
    dataTurbi = analogRead(turbiPin);
    hasilavgTurbi = avgTurbi.reading(dataTurbi);
    voltTurbi = dataTurbi * (5.0 / 1023);
    hasilKeruh = 100.00 - (voltTurbi / 3.86) * 100.00;
    //----- rumus salinitas-----
    dataSalinitas = analogRead(salinitasPin);
    hasilavgSalinitas =
avgSalinitas.reading(dataSalinitas);
    hasilSalinitas = (0.3417 * hasilavgSalinitas) +
110.1 * 12 * 3;

```

```

//-----Rumus sensor ultrasonik-----
digitalWrite(trigPin, LOW);
delayMicroseconds(2);
digitalWrite(trigPin, HIGH);
delayMicroseconds(10);
digitalWrite(trigPin, LOW);
waktu = pulseIn(echoPin, HIGH);
jarak= waktu*0.034/2;
persentase = map(jarak, 50, 5, 0, 100);
Serial.print("pakan: ");
Serial.println(persentase);
Serial.println(" %");
delay(200);
//-----Rumus TEGANGAN-----
adcvolt = analogRead(voltPin);
Vmodul = (adcvolt * 5.0) / 1024.0;
nilaivolt = Vmodul / (R2/(R1+R2));
//-----Rumus Arus-----
adcarus = analogRead(arusPin);
tegangan = (adcarus / 1024.0) * 5000;
nilaiarus = ((tegangan - teganganoffset) /
sensitivitas)*-1;
if (Serial1.available() > 0) {
    dataSerial = Serial1.readStringUntil('\n');
}
int data1 = dataSerial.indexOf('{');
int data2 = dataSerial.indexOf(',', data1 + 1);
int data3 = dataSerial.indexOf(',', data2 + 1);
int data4 = dataSerial.indexOf(',', data3 + 1);
int data5 = dataSerial.indexOf(',', data4 + 1);
int data6 = dataSerial.indexOf(',', data5 + 1);
int data7 = dataSerial.indexOf(',', data6 + 1);
int data8 = dataSerial.indexOf(',', data7 + 1);
int data9 = dataSerial.indexOf(',', data8 + 1);
int data10 = dataSerial.indexOf(',', data9 + 1);
int data11 = dataSerial.indexOf(',', data10 + 1);
int data12 = dataSerial.indexOf(',', data11 + 1);
int data13 = dataSerial.indexOf(',', data12 + 1);

```

```

int data14 = dataSerial.indexOf(',', data13 + 1);
int data15 = dataSerial.indexOf(',', data14 + 1);
int data16 = dataSerial.indexOf('}', data15 + 1);
//pemisah data string
je = dataSerial.substring(data1 + 1, data2);
me = dataSerial.substring(data2 + 1, data3);
de = dataSerial.substring(data3 + 1, data4);

ja = dataSerial.substring(data4 + 1, data5);
ma = dataSerial.substring(data5 + 1, data6);
da = dataSerial.substring(data6 + 1, data7);

js = dataSerial.substring(data7 + 1, data8);
ms = dataSerial.substring(data8 + 1, data9);
ds = dataSerial.substring(data9 + 1, data10);

jm = dataSerial.substring(data10 + 1, data11);
mm = dataSerial.substring(data11 + 1, data12);
dm = dataSerial.substring(data12 + 1, data13);
//data integer hasil jadi
jamPagi = je.toInt();
menitPagi = me.toInt();
detikPagi = de.toInt();

jamSiang = ja.toInt();
menitSiang = ma.toInt();
detikSiang = da.toInt();

jamSore = js.toInt();
menitSore = ms.toInt();
detikSore = ds.toInt();

jamMalam = jm.toInt();
menitMalam = mm.toInt();
detikMalam = dm.toInt();

pakanIn = pakan.toInt();
logicPagi(jamPagi, menitPagi, detikPagi);

```

```

    logicPagi(jamSiang, menitSiang, detikSiang);
    logicPagi(jamSore, menitSore, detikSore);
    logicPagi(jamMalam, menitMalam, detikMalam);
}
void logicPagi(int setJam, int setMenit, int
setDetik) {
if (jam == setJam and menit == setMenit and detik ==
setDetik) {
    countPagi = 1;
}
if (countPagi == 1) {
    if (gram >= pakanIn) {
        countGram = 1;
        openStorage.write(tutup);
    }
    else {
        openStorage.write(buka);//ngisi corong
        openScale.write(tutup);
        delay(10);
    }
    if (countGram == 1) {
        if (gram <= 5 or gram == 0)
        {
            openScale.write(tutup);
            digitalWrite(motor, LOW);
            delay(10);
            countPagi = 0;
            countGram = 0;
        }
        else
        {
            openStorage.write(tutup);
            openScale.write(buka);
            digitalWrite(motor, HIGH);
            delay(5000);
        }
    }
}
}
}

```



```

else {
    openStorage.write(tutup);
}
}
void rtcSet() {
    Rtc.Begin();
    RtcDateTime compiled = RtcDateTime(__DATE__,
__TIME__);
    printDateTime(compiled);
    Serial.println();
    if (!Rtc.IsDateTimeValid())
    {
        if (Rtc.LastError() != 0)
        {
            Serial.print("RTC communications error = ");
            Serial.println(Rtc.LastError());
        }
        else
        {
            Serial.println("RTC lost confidence in the
DateTime!");
            Rtc.SetDateTime(compiled);
        }
    }
    if (!Rtc.GetIsRunning())
    {
        Serial.println("RTC was not actively running,
starting now");
        Rtc.SetIsRunning(true);
    }
    RtcDateTime now = Rtc.GetDateTime();
    if (now < compiled)
    {
        Serial.println("RTC is older than compile
time! (Updating DateTime)");
        Rtc.SetDateTime(compiled);
    }
}

```

```

    Rtc.Enable32kHzPin(false);
    Rtc.SetSquareWavePin(DS3231SquareWavePin_ModeNone);
}
#define countof(a) (sizeof(a) / sizeof(a[0]))
void printDateTime(const RtcDateTime& dt)
{
    char datestring[20];
    snprintf_P(datestring,
               countof(datestring),
               PSTR("%02u/%02u/%04u %02u:%02u:%02u"),
               dt.Month(),
               dt.Day(),
               dt.Year(),
               dt.Hour(),
               dt.Minute(),
               dt.Second() );
    Serial.println(datestring);
}

```

Program ESP32

```

#define BLYNK_PRINT Serial
#include <WiFi.h>
#include <WiFiClient.h>
#include <BlynkSimpleEsp32.h>
#include <SoftwareSerial.h>
unsigned long timeShow ;
byte flagRun ;
const char auth[] =
"YddcpEBibqmacX70Mdxj3Rv8J68lDigz";
const char ssid[] = "sedekah";
const char pass[] = "modalnapa";
String dataMasuk = "";

BlynkTimer timer;
WidgetLCD lcdh(V7);
WidgetLCD lcd(V9);
#define RXD2 16 //hitam

```

```

#define TXD2 17 //putih
float loadcel, ph , turbi , tds, volt, arus;
int cnt = 0;
int tambah , kurang ;
int jam , menit , detik , persentase;
int jam_esuk, menit_esuk, detik_esuk;
int jam_awan, menit_awan, detik_awan;
int jam_sore, menit_sore, detik_sore;
int jam_wengi, menit_wengi, detik_wengi;
//serial Kirim
String jam_esuk_str, menit_esuk_str, detik_esuk_str;
String jam_awan_str, menit_awan_str, detik_awan_str;
String jam_wengi_str, menit_wengi_str,
detik_wengi_str;
String loadcel_str, ph_str , turbi_str , tds_str ,
jam_str , menit_str , detik_str , ultra_str,
volt_str, arus_str;
BLYNK_WRITE(V0) {
    TimeInputParam esuk(param);
    if (esuk.hasStartTime()) {
        jam_esuk = esuk.getStartHour();
        menit_esuk = esuk.getStartMinute();
        detik_esuk = esuk.getStartSecond();
    }
}
BLYNK_WRITE(V1) {
    TimeInputParam awan(param); // read data tombol
    dari blynk
    if (awan.hasStartTime()) {
        jam_awan = awan.getStartHour();
        menit_awan = awan.getStartMinute();
        detik_awan = awan.getStartSecond();
    }
}
BLYNK_WRITE(V2) {
    TimeInputParam sore(param); // read data tombol
    dari blynk
    if (sore.hasStartTime()) {

```

```

    jam_sore = sore .getStartHour();
    menit_sore = sore .getStartMinute();
    detik_sore = sore.getStartSecond();
}
}
BLYNK_WRITE(V5) {
    TimeInputParam wengi(param);// read data tombol
    dari blynk
    if (wengi.hasStartTime()) {
        jam_wengi = wengi .getStartHour();
        menit_wengi = wengi .getStartMinute();
        detik_wengi = wengi.getStartSecond();
    }
}
//////////set abot//////////
BLYNK_WRITE(V3) {
    int tambah = param.asInt();
    Serial.println(tambah);
    if (tambah == 1 ) {
        cnt += 125;
    }
}
BLYNK_WRITE(V4) {
    int kurang = param.asInt();
    if (kurang == 1) {
        cnt -= 125;
    }
}
void myTimerEvent()
{
    show();//fungsi menampilkan lcd pada blynk
}
void setup() {
    // open serial for monitoring
    Serial.begin(9600);
    // Serial2.begin(9600);
    Blynk.begin(auth, ssid, pass, "blynk.cloud", 80);
    Serial2.begin(9600);
}

```

```

    timer.setInterval(1000L, myTimerEvent);
}
void loop() {
    serial_terima();
    Serial.println(dataMasuk);
    Serial.print("cnt =");
    Serial.print(cnt);
    kirim();
    Blynk.run();
    timer.run();
    Blynk.virtualWrite(V6 , cnt );
    sendSensor();
}
//////////////////////////////////TERIMA DATA//////////////////////////////////
void serial_terima() {

    if (Serial2.available() > 0) {
        dataMasuk = Serial2.readStringUntil('\n' );
        dataMasuk.trim();

        byte x1 = dataMasuk.indexOf('{');
        byte x2 = dataMasuk.indexOf(',', x1 + 1);
        byte x3 = dataMasuk.indexOf(',', x2 + 1);
        byte x4 = dataMasuk.indexOf(',', x3 + 1);
        byte x5 = dataMasuk.indexOf(',', x4 + 1);
        byte x6 = dataMasuk.indexOf(',', x5 + 1);
        byte x7 = dataMasuk.indexOf(',', x6 + 1);
        byte x8 = dataMasuk.indexOf(',', x7 + 1);
        byte x9 = dataMasuk.indexOf(',', x8 + 1);
        byte x10 = dataMasuk.indexOf(',', x9 + 1);
        byte x11 = dataMasuk.indexOf('}', x10 + 1);
        loadcel_str = dataMasuk.substring(x1 + 1, x2);
        ph_str = dataMasuk.substring(x2 + 1, x3);
        turbid_str = dataMasuk.substring(x3 + 1, x4);
        tds_str = dataMasuk.substring(x4 + 1, x5);
        jam_str = dataMasuk.substring(x5 + 1, x6);
        menit_str = dataMasuk.substring(x6 + 1, x7);
        detik_str = dataMasuk.substring(x7 + 1, x8);
    }
}

```

```

    ultra_str = dataMasuk.substring(x8 + 1, x9);
    arus_str = dataMasuk.substring(x9 + 1, x10);
    volt_str = dataMasuk.substring(x10 + 1, x11);

    loadcel = loadcel_str.toFloat();
    arus = arus_str.toFloat();
    volt = volt_str.toFloat();
    ph = ph_str.toFloat();
    turbi = turbi_str.toFloat();
    tds = tds_str.toFloat();
    persentase = ultra_str.toFloat();
    jam = jam_str.toInt();
    menit = menit_str.toInt();
    detik = detik_str.toInt();
}
}
////////////////////////////////LCD////////////////////////////////////
void show() {
    if (millis() - timeShow > 3000) {lcdh.clear();
        lcdh.print(0, 0, " MONITORING DATA ");
        flagRun++;
        timeShow = millis();
    }
    switch (flagRun) {
        case 0:
            break;
        case 1:
            lcdh.print(0, 1, " BERAT:      gram" );
            lcdh.print(7, 1, String (loadcel));
            break;
        case 2:
            lcdh.print(0, 1, " PH :      ");
            lcdh.print(6, 1, String(ph));
            break;
        case 3:
            lcdh.print(0, 1, " TURBI:      NTU");
            lcdh.print(7, 1, String (turbi));
            break;
    }
}

```

```

    case 4:
        lcdh.print(0, 1, "Salin:          PPM");
        lcdh.print(6, 1, String (tds));
        break;
    case 5:
        lcdh.print(0, 1, "Volt :          V");
        lcdh.print(6, 1, String (volt));
        break;
    case 6:
        lcdh.print(0, 1, "Arus:          A");
        lcdh.print(6, 1, String (arus));
        break;
    case 7:
        flagRun = 1;
        break;
}
}
void kirim() {
    Serial2.print("{");
    Serial2.print(jam_esuk);
    Serial2.print(",");
    Serial2.print(menit_esuk);
    Serial2.print(",");
    Serial2.print(detik_esuk);
    Serial2.print(",");
    Serial2.print(jam_awan);
    Serial2.print(",");
    Serial2.print(menit_awan);
    Serial2.print(",");
    Serial2.print(detik_awan);
    Serial2.print(",");
    Serial2.print(jam_sore);
    Serial2.print(",");
    Serial2.print(menit_sore);
    Serial2.print(",");
    Serial2.print(detik_sore);
    Serial2.print(",");
    Serial2.print(jam_wengi);
}

```

```

Serial2.print(",");
Serial2.print(menit_wengi);
Serial2.print(",");
Serial2.print(detik_wengi);
Serial2.print(",");
Serial2.print(cnt);
Serial2.println("}");
Serial.print("JAM_esuk =" + String(jam_esuk) + ":"
+ String(menit_esuk) + ":" + String(detik_esuk));
Serial.print("JAM_awan =" + String(jam_awan) + ":"
+ String(menit_awan) + ":" + String(detik_awan));
Serial.print("JAM_sore =" + String(jam_sore) + ":"
+ String(menit_sore) + ":" + String(detik_sore));
Serial.println("JAM_wengi =" + String(jam_wengi) +
":" + String(menit_wengi) + ":" +
String(detik_wengi));
}
void sendSensor(){
    Blynk.virtualWrite(V8, persentase);
}

```


LAMPIRAN B

Tampilan Alat



Tampak Samping



Tampak Depan



Tampak Belakang

BIODATA PENULIS



Nama : Nur Afif Hidayat Mustofa
Tempat/Tanggal Lahir : Cilacap, 10 November 2000
Alamat : Desa Danasri Lor RT002 RW013
Kec.Nusawungu , Kab.Cilacap (53283)
Email : afifmustofa1011@gmail.com
Telepon/HP : 088224107453
Hobi : Menyanyi
Motto : Ambeg utomo, andhap asor

Riwayat Pendidikan

- SDN Danasri Lor 03 Tahun 2007-2013
- SMP Pemda Nusawungu Tahun 2013-2016
- MAN 3 Banyumas
Jurusan IPA Tahun 2017-2020
- Politeknik Negeri Cilacap
Prodi D3 Teknik Elektronika Tahun 2020-2023

Penulis telah mengikuti sidang Tugas pada tanggal 11 Agustus 2023 sebagai salah satu persyaratan untuk memperoleh gelar Ahli Madya (A.Md)