DAFTAR PUSTAKA

Ahman, E., & Indriani, E. (2007). *Ekonomi dan Akuntansi: Membina Kompetensi Ekonomi*. Bandung: PT Grafindo Media Pratama.

Aldrianto, A. Y. (2015). Rancang Otomasi Kontrol Temperatur dan Ph Air Pada Kebun Hidroponik Tanaman Selada Keriting. Surabaya: Institut Teknologi Sepuluh Nopember.

Bafdal, N., & Ardiansah, I. (2020). *Smart Farming Berbasis Internet of Things dalam Greenhouse.* Sumedang: Unpad Press.

Blynk. (2021). *Blynk Docs*. Diambil kembali dari Blynk: https://docs.blynk.cc/#intro-how-blynk-works

Darmawan, I. A. (1997). *Pengaruh Topoklimat terhadap Produksi dan Kualitas Selada (Lactuca sativa L.).* Yogyakarta: Universitas Gadjah Mada.

Emma, S. (2021, Mei 31). *Cara Mengatasi Gangguan Budidaya Tanaman Hidroponik*. Diambil kembali dari Balai Pengkajian Teknologi Pertanian Sumatera Selatan: https://sumsel.litbang.pertanian.go.id/berita-cara-mengatasi-gangguan-budidaya-tanaman-hidroponik.html

Firdaus, Z. (2009). *Korelasi Antara Pelatihan Teknis Perpajakan, Pengalaman dan Motivasi Pemeriksa Pajak Dengan Kinerja Pemeriksa Pajak Pada Kantor Pelayanan Pajak Di Jakarta Barat.* Jakarta: Universitas Islam Negeri Syarif Hidayatullah.

Fitrya, N., Ginting, D., Retnawaty, S. F., Fitri, Y., & Wirman, S. P. (2017). Pentingnya Akurasi dan Presisi Alat Ukur Dalam Rumah Tangga. *Jurnal Untuk Mu negeRI Vol. 1*, 61-64.

Harsanto, B. (2020). Informatika Pertanian. *Inovasi Internet of Things Pada Sektor Pertanian : Pendekatan Analisis Scientometrics*, 112.

Haryanto, B., Ismail, N., & Joni, E. P. (2018). Sistem Monitoring Suhu dan Kelembaban Secara Nirkabel pada Budidaya Tanaman Hidroponik. *Jurnal Teknologi Rekayasa, Vol. 3, No. 1*, 47-54.

Haryanto, E., Suhartini, T., & Rahayu, E. (2003). *Sawi dan Selada*. Jakarta: Penebar Swadaya.

Irwansyah. (2018). *Pemantauan Gas Beracun Dan Debu Dari Asap Kendaraan Bermotor Menggunakan Teknologi Wireless Sensor Network dan Regresi Linier.* Medan: Universitas Sumatera Utara.

Jones, L. D., & A, F. C. (1991). *Electronic Instruments and Measurements 2nd Edition.* Pearson College Div; 2nd edition.

Komaludin, D. (2018). Penerapan Teknologi Internet of Thing (IoT) Pada Bisnis Budidaya Tanaman Hidroponik Sebagai Langkah Efisiensi Biaya Perawatan. *Festival Riset Ilmiah Manajemen & Akuntansi*, 684-686.

Kuncoro, B. D., Sutandi, T., & Falahuddin, A. (2017). Pengembangan Sistem Pendingin Larutan Nutrisi untuk Budidaya Tanaman Hidroponik. *Saintiks-2-2017*, 67-72.

Kurniawan, D. (2008). *REGRESI LINIER*. Diambil kembali dari FORUM STATISTIKA: https://ineddeni.wordpress.com/

Nazaruddin. (2003). *Budidaya dan Pengaturan Panen Sayuran Dataran Rendah.* Jakarta: Penebar Swadaya.

Nugroho, B. A. (2005). *Strategi Jitu Memilih Metode Statistik Penelitian dengan SPSS.* Yogyakarta: Andi.

Puspita S, D. A. (2017). *Rancang Bangun Alat Ukur Temperatur Ruangan dengan Memanfaatkan Sensor DHT22 Berbasis Arduino UNO.* Medan: Universitas Sumatera Utara.

Putera, G. A., & D, C. (2017). Perancang Alat Ukur Kadar Padatan Terlarut, Kekeruhan dan Ph Air Menggunakan Arduino Uno. Makassar: Universitas Hasanuddin.

Qurrohman, B. F. (2019). *Bertanam Selada Hidroponik Konsep dan Aplikasi.* Bandung: Pusat Penelitian dan Penerbitan UIN SGD Bandung.

Roberto, K. (2003). *How-To Hydroponics Fourth Edition.* The Futuregarden Press.

Rudy, G., Andhika, T., Sandi, & Hibatullah, F. (2019). Sistem Monitoring Kelembaban Tanah, Suhu, pH dan Penyiraman Otomatis Pada Tanaman Tomat Berbasis Internet of Things. *Telekontran, Vol 7, No 1*, 74.

Ruengittinun, S., Phongsamsuan, S., & Sureeratanakorn, P. (2017). Applied Internet of Things for Smart Hydroponic Farming Ecosystem (HFE). *10th*

International Conference on Ubi-media Computing and Workshops (Ubi-Media), 4.

Saputra, I., Triyanto, D., & Ruslianto, I. (2015). Sistem Kendali Suhu, Kelembaban, dan Level Air Pada Pertanian Pola Hidroponik. *Jurnal Coding, Sistem Komputer Untan Volume 03, No. 1*, 1-10.

Setiawati, I., & Harsono, B. (2020). Sistem Hidroponik Berbasis Internet of Things. *Dielektrika, 7 (2)*, 82-97.

Sudarmo, A. P. (2018). Pemanfaatan Pertanian Secara Hidroponik Untuk Mengatasi Keterbatasan Lahan Pertanian Di Daerah Perkotaan. *Seminar Nasional Pengabdian Kepada Masyarakat Universitas Terbuka*, 1-8.

Susila, A. D. (2013). *Bahan Ajar Mata Kuliah Dasar-Dasar Hortikultura* . Bogor: Institut Pertanian Bogor.

Tallei, T. E., Rumengan, I. F., & Adam, A. A. (2017). *Hidroponik Untuk Pemula.* Manado: LPPM UNSRAT.

Tinton, D. P. (2015). *Hidroponik Wick System: Cara Paling Praktis, Pasti Panen.* Jakarta: AgroMedia.

Union, I. T. (2012). Overview of the Internet of things. *Recommendation ITU-T Y.2060*, 2.

Untung, O. (2004). *Hidroponik sayuran sistem NFT(Nutrient film technique).* Jakarta: Penebar Swadaya.

Wahome, P. K., Oseni, T. O., Masarirambi, M. T., & Shongwe, V. D. (2011). Effects of Different Hydroponics Systems and Growing Media on the Vegetative Growth, Yield and Cut Flower Quality of Gypsophila (Gypsophila paniculata L.). *World Journal of Agricultural Sciences 7 (6)*, 692-698.

Yustiningsih, M., Naisumu, Y. G., & Berek, A. (2016). Deep Flow Technique (Dft) Hidroponik Menggunakan Media Nutrisi Limbah Cair Tahu Dan Kayu Apu (Pistia Stratiotes L) Untuk Peningkatan Produktivitas Tanaman. *Biologi dan Pendidikan Biologi*, 111.

LAMPIRAN

Lampiran 1. *Source Code*

```
//-----DEKLARASI VARIABEL DAN FUNGSI-----//
#define BLYNK_PRINT Serial
#include <ESP8266_Lib.h>
#include <BlynkSimpleShieldEsp8266.h>
#include "DHT.h"
#include "NewPing.h"
#include <OneWire.h>
#include <DallasTemperature.h>
#include <WidgetRTC.h>
#include <EEPROM.h>
#include "GravityTDS.h"

char auth[] = "n-afJTn4zEvmb9yxR-Co7Z3eZ-4RfUy1";
char ssid[] = "KOS_FLOWERS";
char pass[] = "2701bunga";
char Date[16];
char Time[16];
float distance;
float tdsValue = 0;
float nilai_tds;
float ds18b20Cal;
const int phSensorPin = A0;
int phSensorValue = 0;
unsigned long int phAvgValue;
float phValue;
float pHVol;
int buf[10];
int temp;
bool isFirstConnect = true;
int manual ;
int pinCheckStatus ;
int pinPhMode ;
int pinFanMode ;
int pinUltrasonic ;
int AutoCirculation ;
long startsecondswd;              // weekday start time in seconds
long stopsecondswd;               // weekday stop  time in seconds
long nowseconds;                  // time now in seconds
float t;
float CalH;
float CalT;

#define EspSerial Serial1
#define ESP8266_BAUD 9600
ESP8266 wifi(&EspSerial);

#define DHTPIN 2
#define DHTTYPE DHT22
#define TRIGGER_PIN 4
#define ECHO_PIN 5
#define MAX_DISTANCE 400
#define ONE_WIRE_BUS 3
#define TdsSensorPin A1
```

```
#define RelayCirculation 6
#define RelayUltraSonic 7
#define RelayFan 8
#define  RelayPhUp 10
#define RelayPhDown 11


NewPing sonar(TRIGGER_PIN, ECHO_PIN, MAX_DISTANCE);
GravityTDS gravityTds;
DHT dht(DHTPIN, DHTTYPE);
BlynkTimer timer;
WidgetTerminal terminal(V1);
WidgetRTC rtc;
WidgetLED led1(V5);
WidgetLED led2(V6);
WidgetLED led3(V27);
WidgetLED led4(V28);

OneWire oneWire(ONE_WIRE_BUS);
DallasTemperature  ds18b20(&oneWire);

void setup(){
    // Debug console
    Serial.begin(9600);
    EspSerial.begin(ESP8266_BAUD);
    delay(10);

    Blynk.begin(auth, wifi, ssid, pass);

    ds18b20.begin();  // Start up the library
    rtc.begin();
    dht.begin();

    gravityTds.setPin(TdsSensorPin);
    gravityTds.setAref(5.0);
    gravityTds.setAdcRange(1024);
    gravityTds.begin();  //initialization

    pinMode(RelayCirculation, OUTPUT);
    pinMode(RelayUltraSonic, OUTPUT);
    pinMode(RelayFan, OUTPUT);
    pinMode(RelayPhUp, OUTPUT);
    pinMode(RelayPhDown, OUTPUT);

    digitalWrite(RelayCirculation, LOW);
    digitalWrite(RelayUltraSonic, LOW);
    digitalWrite(RelayFan, LOW);
    digitalWrite(RelayPhUp, LOW);
    digitalWrite(RelayPhDown,LOW);


    timer.setInterval(1000L, sendSensor); // 1 SEC
    timer.setInterval(60000L, phUpUpdateValue);
    timer.setInterval(1000L,UltrasonicUpdateValue);
    timer.setInterval(1000L,  FanUpdateValue);
    timer.setInterval(60000L, reconnectBlynk); // 30 SEC
    timer.setInterval(10000L, activetoday); // 10 SEC
}
```

```
BLYNK_CONNECTED() {
    if (isFirstConnect) {
        Blynk.syncAll();
        Blynk.notify("TIMER STARTING!!!!");
        isFirstConnect = false;
    }
}

void sendSensor() {
//-----------------------DHT SENSOR -------------------------//
    t = dht.readTemperature();
    float CalT =  (t - 2.8154 )/ 0.9465 ;// y = 0,9465x + 2,8154
    Blynk.virtualWrite(V10, CalT);

//---------------------- DS18B20 Sensor ----------------------//
    ds18b20.requestTemperatures();
    ds18b20Cal = (ds18b20.getTempCByIndex(0) - 2.7321) / 0.9704;
    Blynk.virtualWrite(V11, ds18b20Cal);

//---------------------- HYSRF-05 SENSOR ----------------------//
    distance = sonar.ping_cm();
    Blynk.virtualWrite(V14, distance);

//----------------------PH CALIBRATION ----------------------//

    for(int i=0;i<10;i++) {
        buf[i]=analogRead(phSensorPin);
        delay(10);
    }

    for(int i=0;i<9;i++) {
        for(int j=i+1;j<10;j++) {
            if(buf[i]>buf[j]) {
                temp=buf[i];
                buf[i]=buf[j];
                buf[j]=temp;
            }
        }
    }

    phAvgValue=0;
    for(int i=2;i<8;i++)
    phAvgValue+=buf[i];
    pHVol=(float)phAvgValue*5.0/1024/6;
    phValue =  5,685*(pHVol) + 20,403;//-0,1759x + 3,5887
    Blynk.virtualWrite(V16,pHVol);
    Blynk.virtualWrite(V12,phValue);

//--------------------- TDS CALIBRATION ----------------------//

    gravityTds.setTemperature(ds18b20Cal);
    gravityTds.update();
    tdsValue = gravityTds.getTdsValue();
    float voltage = TdsSensorPin * (5.0 / 1024.0);
    float nilai_tds = (0,8382 - tdsValue) /(-0,0015);
    Blynk.virtualWrite(V19,voltage);
    Blynk.virtualWrite(V18,nilai_tds);

}
```

```
//---------------------- Check Status  ----------------------//

BLYNK_WRITE(V26) {  // Check Status Sensor
  if (param.asInt()==1) {
    Blynk.virtualWrite(V1, "clr");
    Blynk.virtualWrite(V1,"\n Mengambil Nilai Sensor...");
    Blynk.virtualWrite(V1, "\n");
    Blynk.virtualWrite(V1, "\n");
    led1.off();
    led2.off();
    led3.off();
    led4.off();
    pinCheckStatus = 1;
    pinPhMode = 0;
    pinFanMode = 0;
    AutoCirculation=0;
    pinUltrasonic = 0;
    Blynk.virtualWrite(V29, 0);
    Blynk.virtualWrite(V2, 0);
    Blynk.virtualWrite(V4, 0);   //Turn ON Button Widge
    Blynk.virtualWrite(V26, 1);
    Blynk.virtualWrite(V24, 0);
    Blynk.virtualWrite(V20, 0);
    Blynk.virtualWrite(V30, 0);
    Blynk.virtualWrite(V31, 0);

//----------------------- CHECK DHT  -------------------------//
    t = dht.readTemperature();
    float CalT =  (t - 2.8154 )/ 0.9465 ;// y = 0,9465x + 2,8154
    Blynk.virtualWrite(V1, "clr");
    Blynk.virtualWrite(V1, "\n");
    Blynk.virtualWrite(V1, "\n Suhu Lingkungan : ", CalT,"°C");
        if (CalT >= 26 && CalT<=31 ){
          Blynk.virtualWrite(V1, "\n Status suhu :  Normal");
        }
        else if (CalT >30){
          Blynk.virtualWrite(V1, "\n Status suhu :  Panas");
         }
        else if (CalT < 26){
            Blynk.virtualWrite(V1, "\n Status suhu :  Dingin");
        }
        else {
            Blynk.virtualWrite(V1, "Sensor Tidak Terbaca");
        }

//---------------------- CHECK DS18B20  ----------------------//
    ds18b20.requestTemperatures();
    ds18b20Cal = (ds18b20.getTempCByIndex(0) - 2.7321) / 0.9704 ;
//Y=0,9704x + 2,7321
    Blynk.virtualWrite(V1, "\n ");  // Clear the terminal from
past info.
    Blynk.virtualWrite(V1, "\n ");
    Blynk.virtualWrite(V1, "\n Suhu Air : ",  ds18b20Cal,"°C");
        if (ds18b20Cal >= 25 &&  ds18b20Cal<=27 ){
            Blynk.virtualWrite(V1, "\n Status suhu Air :
Normal");
        }
        else if ( ds18b20Cal >27){
```

```
            Blynk.virtualWrite(V1, "\n Status suhu Air :  Panas");
        }
        else if ( ds18b20Cal < 25){
            Blynk.virtualWrite(V1, "\n Status suhu Air :
Dingin");
        }
        else {
            Blynk.virtualWrite(V1, "Sensor Tidak Terbaca");
        }

//---------------- CHECK HC-SR04 UltraSonic  -----------------//

    Blynk.virtualWrite(V1, "\n ");
    Blynk.virtualWrite(V1, "\n ");
    Blynk.virtualWrite(V1, "\n Kondisi Air Pada Bak Penampungan :
", distance ,"Cm");
        if (distance >= 3 && distance <= 7 ){
            Blynk.virtualWrite(V1, "\n Status Bak Penampungan :
Normal");
        }
        else if (distance > 7 ){
            Blynk.virtualWrite(V1, "\n Status Bak Penampungan :
Hampir Habis + Nyalakan Pompa");
        }
        else if (distance <= 3){
            Blynk.virtualWrite(V1, "\n Status Bak Penampungan :
Full  + Pompa Mati");
        }
        else {
            Blynk.virtualWrite(V1, "Sensor Tidak Terbaca");
        }

//------------------------ CHECK pH  -------------------------//

    Blynk.virtualWrite(V1, "\n ");
    Blynk.virtualWrite(V1, "\n ");
    Blynk.virtualWrite(V1, "\n pH Air : ", phValue ," pH");
        if (phValue >= 6 && phValue <= 7 ){
            Blynk.virtualWrite(V1, "\n Status pH Air : pH
Normal");
            Blynk.virtualWrite(V1, "\n Status Pompa pH Up & Down :
Tidak Aktif");
        }
        else if (phValue  > 7){
            Blynk.virtualWrite(V1, "\n Status pH Air :  pH Terlalu
Tiggi ( Basa )");
            Blynk.virtualWrite(V1, "\n Status Pompa pH Down :
Aktif ");
        }
        else if (phValue < 6){
            Blynk.virtualWrite(V1, "\n Status pH Air :  pH Terlalu
Rendah ( Asam )");
            Blynk.virtualWrite(V1, "\n Status Pompa pH Up :
Aktif");
        }
        else {
            Blynk.virtualWrite(V1, "Sensor Tidak Terbaca");
        }
```

```
 //----------------------- CHECK TDS  ------------------------//

    gravityTds.setTemperature(ds18b20Cal);
    gravityTds.update();
    tdsValue = gravityTds.getTdsValue();
    float voltage = TdsSensorPin * (5.0 / 1024.0);
    float nilai_tds = (0,8382 - tdsValue) /(-0,0015);

    Blynk.virtualWrite(V1, "\n ");
    Blynk.virtualWrite(V1, "\n ");
    Blynk.virtualWrite(V1, "\n Kandungan Nutrisi Tanaman : ",
nilai_tds ,"ppm ");
        if (nilai_tds > 840){
            Blynk.virtualWrite(V1, "\n Status TDS Air :  Nutrisi
Berlebih Pada Air = TDS TIdak Sesuai");
        }
        else if (nilai_tds < 560 ){
            Blynk.virtualWrite(V1, "\n Status TDS Air :  Nutrisi
Kurang Pada Air = TDS TIdak Sesuai");
        }
        else {
            Blynk.virtualWrite(V1, " (Nurisi Normal)");
        }
  }
   else {
      Blynk.virtualWrite(V1, "clr");
      Blynk.virtualWrite(V1, "\n");
      Blynk.virtualWrite(V1,"\n Mode Check Status Tanaman  OFF ");
      Blynk.virtualWrite(V1,"\n Mode OFF");
      Blynk.virtualWrite(V1, "\n");
      pinCheckStatus = 0;
      pinPhMode = 0;
      pinFanMode = 0;
      AutoCirculation=0;
      Blynk.virtualWrite(V2, 0);
      Blynk.virtualWrite(V4, 0);   //Turn OFF Button Widget
      Blynk.virtualWrite(V26, 0);
      Blynk.virtualWrite(V24, 0);
      Blynk.virtualWrite(V20, 0);
      Blynk.virtualWrite(V30, 0);
      Blynk.virtualWrite(V31, 0);
      led2.off();
    }
}

//------------------ PENYIRAMAN MANUAL  ---------------------//

BLYNK_WRITE(V4){
    if (param.asInt()==1){
        Blynk.virtualWrite(V1, "clr");
        Blynk.virtualWrite(V1, "\n");
        Blynk.virtualWrite(V1, "\n Manual Mode is ON");
        Blynk.virtualWrite(V1, "\n");
        led1.off();
        led2.on();
        led3.off();
        led4.off();

        pinCheckStatus = 0;
```

```
        pinPhMode = 0;
        pinFanMode = 0;
        AutoCirculation=0;
        pinUltrasonic = 0;
        Blynk.virtualWrite(V29, 0);
        Blynk.virtualWrite(V2, 0);
        Blynk.virtualWrite(V4, 1);   //Turn ON Button Widge
        Blynk.virtualWrite(V26, 0);
        Blynk.virtualWrite(V24, 0);
        digitalWrite(RelayCirculation, HIGH); // set LED ON
    }

    else {
        Blynk.virtualWrite(V1, "clr");
        Blynk.virtualWrite(V1, "\n");
        Blynk.virtualWrite(V1,"\n Mode Manual  OFF ");
        Blynk.virtualWrite(V1, "\n");
        pinCheckStatus = 0;
        pinPhMode = 0;
        pinFanMode = 0;
        AutoCirculation=0;
        pinUltrasonic = 0;
        Blynk.virtualWrite(V29, 0);
        Blynk.virtualWrite(V2, 0);
        Blynk.virtualWrite(V4, 0);   //Turn OFF Button Widget
        Blynk.virtualWrite(V26, 0);
        Blynk.virtualWrite(V24, 0);
        Blynk.virtualWrite(V20, 0);
        digitalWrite(RelayCirculation, LOW); // set LED OFF
        led1.off();
        led2.off();
    }
}

//----------------------- KONTROL PH  ------------------------//

void phUpUpdateValue () {
    if(phValue > 7){
        if (pinPhMode==1) {
            Blynk.syncVirtual(V24);
        }
    }

    else if (phValue < 6){
        if (pinPhMode==1) {
            Blynk.syncVirtual(V24);
        }
    }
}

BLYNK_WRITE(V24){
    if (param.asInt()==1) {
        Blynk.virtualWrite(V1, "clr");
        Blynk.virtualWrite(V1, "\n");
        Blynk.virtualWrite(V1,"\n Kontrol pH Aktif ");
        Blynk.virtualWrite(V1,"\n Mengambil data pH...");
        Blynk.virtualWrite(V1, "clr");
        Blynk.virtualWrite(V1, "\n ");
        Blynk.virtualWrite(V1, "\n ");
```

```
        Blynk.virtualWrite(V1, "\n pH Air : ", phValue ," pH");
        Blynk.virtualWrite(V1, "\n");
        Blynk.virtualWrite(V24, 1);
        led1.off();
        led2.off();
        led3.on();
        led4.off();
        pinCheckStatus = 0;
        pinPhMode = 1;
        pinFanMode = 0;
        AutoCirculation=0;
        Blynk.virtualWrite(V2, 0);
        Blynk.virtualWrite(V4, 0);
        Blynk.virtualWrite(V24, 1);
        Blynk.virtualWrite(V26, 0);

            if (phValue >= 6 && phValue <=7){
                Blynk.virtualWrite(V1, "\n Status pH Air : pH
Normal");
                Blynk.virtualWrite(V1, "\n Status Pompa pH Up &
Down : Tidak Aktif");
                Blynk.virtualWrite(V1, "\n");
                digitalWrite(RelayPhDown, LOW);
                digitalWrite(RelayPhUp, LOW);
            }

            else if (phValue < 6){
                Blynk.virtualWrite(V1, "\n Status pH Air :  pH
Terlalu Rendah ( Asam )");
                Blynk.virtualWrite(V1, "\n Status Pompa pH Up :
Aktif");
                Blynk.virtualWrite(V1, "\n");
                digitalWrite(RelayPhUp, HIGH);
                delay(300);
                digitalWrite(RelayPhUp, LOW);
            }

            else if (phValue  > 7  ){
                Blynk.virtualWrite(V1, "\n Status pH Air :  pH
Terlalu Tiggi ( Basa )");
                Blynk.virtualWrite(V1, "\n Status Pompa pH Down :
Aktif");
                Blynk.virtualWrite(V1, "\n");
                digitalWrite(RelayPhDown, HIGH);
                delay(300);
                digitalWrite(RelayPhDown, LOW);
            }
    }

    else {
        Blynk.virtualWrite(V1, "clr");
        Blynk.virtualWrite(V1, "\n ");
        Blynk.virtualWrite(V1, "\n Kontrol pH OFF ");
        Blynk.virtualWrite(V1, "\n ");
        led3.off();
        pinPhMode = 0;
        Blynk.virtualWrite(V24, 0);
        digitalWrite(RelayPhUp, LOW);
        digitalWrite(RelayPhDown, LOW);
```

```
    }
}


//-------------------- UltrasoniicMode  ---------------------//

void UltrasonicUpdateValue () {
    if(distance > 7){
        if (pinUltrasonic==1) {
            Blynk.syncVirtual(V29);
        }
    }

    else if (distance <= 3){
        if (pinUltrasonic==1) {
            Blynk.syncVirtual(V29);
        }
    }
}

//---------------------- Ultrasonic MODE ----------------------//

BLYNK_WRITE(V29) {
    if (param.asInt()==1) {
        Blynk.virtualWrite(V1, "clr");  // Clear the terminal from
past info.
        Blynk.virtualWrite(V1, "\n");
        Blynk.virtualWrite(V1,"\n Mode Ultrasonic Aktif.... ");
        led1.off();
        led2.off();
        led3.off();
        led4.off();
        Blynk.virtualWrite(V1, "\n ");
        Blynk.virtualWrite(V1, "clr");
        pinCheckStatus = 0;
        pinPhMode = 0;
        pinFanMode = 0;
        pinUltrasonic = 1;
        AutoCirculation=0;
        Blynk.virtualWrite(V2, 0);
        Blynk.virtualWrite(V4, 0);
        Blynk.virtualWrite(V26, 0);
        Blynk.virtualWrite(V29, 1);

         if ( distance >= 3 && distance <= 7 ){
            Blynk.virtualWrite(V1, "\n ");
            Blynk.virtualWrite(V1, "\n Jarak Air dan Sensor :
",distance ," °Cm");
            Blynk.virtualWrite(V1, "\n Persedian Air =  Normal");
            Blynk.virtualWrite(V1, "\n Status Pompa = Tidak Aktif
");
            Blynk.virtualWrite(V1, "\n");
            Blynk.virtualWrite(V1,"\n >> Press ON/OFF button if
required <<");
            Blynk.virtualWrite(V1, "\n");
            Blynk.virtualWrite(V1, "\n");
          }

          else if (distance<= 3){
```

```
            Blynk.virtualWrite(V1, "\n ");
            Blynk.virtualWrite(V1, "\n Jarak Air dan Sensor :
",distance ," °Cm");
            Blynk.virtualWrite(V1, "\n Persedian Air =  Penuh");
            Blynk.virtualWrite(V1, "\n Status Pompa = Tidak Aktif
");
            Blynk.virtualWrite(V1, "\n");
            digitalWrite(RelayUltraSonic, LOW);
          }

          else if (distance > 7){
            Blynk.virtualWrite(V1, "\n ");
            Blynk.virtualWrite(V1, "\n Jarak Air dan Sensor :
",distance ," °Cm");
            Blynk.virtualWrite(V1, "\n Persedian Air =  Hampir
Habisl");
            Blynk.virtualWrite(V1, "\n Status Pompa = Menyala");
            Blynk.virtualWrite(V1, "\n");
            digitalWrite(RelayUltraSonic,HIGH);
          }
    }

    else {
        Blynk.virtualWrite(V1, "clr");
        Blynk.virtualWrite(V1, "\n");
        Blynk.virtualWrite(V1,"\n Ultrasonic  MODE is OFF ");
        Blynk.virtualWrite(V1,"\n Device is OFF");
        Blynk.virtualWrite(V1, "\n");
        pinUltrasonic = 0;
        digitalWrite(RelayUltraSonic, LOW);
        Blynk.virtualWrite(V29, 0);
    }
}

//----------------------- FAN Mode ------------------------//

void FanUpdateValue () {
    if(CalT > 28){
        if (pinFanMode==1) {
            Blynk.syncVirtual(V20);
        }
    }

    else if (CalT <25){
        if (pinFanMode==1) {
            Blynk.syncVirtual(V20);
        }
    }
}

//---------------------- KONTROL SUHU ----------------------/

BLYNK_WRITE(V20) {
    if (param.asInt()==1) {
        t = dht.readTemperature();
        float CalT =  (t - 2.8154 )/ 0.9465 ;// y = 0,9465x +
2,8154
        Blynk.virtualWrite(V1, "clr");
        Blynk.virtualWrite(V1, "\n");
```

```
        Blynk.virtualWrite(V1,"\n FAN MODE ACTIVE.... ");
        led1.off();
        led2.off();
        led3.off();
        led4.on();
        Blynk.virtualWrite(V1, "clr");
        pinCheckStatus = 0;
        pinPhMode = 0;
        AutoCirculation=0;
        Blynk.virtualWrite(V2, 0);
        Blynk.virtualWrite(V4, 0);
        Blynk.virtualWrite(V24, 0);
        Blynk.virtualWrite(V26, 0);
        Blynk.virtualWrite(V30, 0);
        Blynk.virtualWrite(V31, 0);

        if (CalT >= 25 && CalT <=28 ){
            Blynk.virtualWrite(V1, "\n ");
            Blynk.virtualWrite(V1, "\n Suhu Lingkungan : ", CalT
," °C");
            Blynk.virtualWrite(V1, "\n Kondisi Suhu Lingkungan =
Normal");
            Blynk.virtualWrite(V1, "\n Status Kipas = Tidak Aktif
");
            Blynk.virtualWrite(V1, "\n");
            digitalWrite(RelayFan, LOW);
        }

        else if (t < 25){
            Blynk.virtualWrite(V1, "\n ");
            Blynk.virtualWrite(V1, "\n Suhu Lingkungan : ", CalT
," °C");
            Blynk.virtualWrite(V1, "\n Kelembapan : ",  CalH ,
"%");
            Blynk.virtualWrite(V1, "\n Kondisi Suhu Lingkungan =
Terlalu Dingin");
            Blynk.virtualWrite(V1, "\n Status Kipas = Tidak Aktif
");
            Blynk.virtualWrite(V1, "\n");
            digitalWrite(RelayFan, LOW);
        }

        else if (t  > 28){
            Blynk.virtualWrite(V1, "\n ");
            Blynk.virtualWrite(V1, "\n Suhu Lingkungan : ", CalT,"
°C");
            Blynk.virtualWrite(V1, "\n Kelembapan : ",  CalH ,
"%");
            Blynk.virtualWrite(V1, "\n Kondisi Suhu Lingkungan =
Terlalu Panas");
            Blynk.virtualWrite(V1, "\n Status Kipas = Aktif");
            Blynk.virtualWrite(V1, "\n");
            digitalWrite(RelayFan, HIGH);
        }
    }

    else {
        Blynk.virtualWrite(V1, "clr");
        Blynk.virtualWrite(V1, "\n");
```

```
        Blynk.virtualWrite(V1,"\n FAN MODE is OFF ");
        Blynk.virtualWrite(V1,"\n Device is OFF");
        Blynk.virtualWrite(V1, "\n");
        led4.off();
        pinFanMode = 0;
        digitalWrite(RelayFan, LOW);
        Blynk.virtualWrite(V20, 0);
    }
}

//-------------------- AUTO CIRCULATION ----------------------//

void activetoday(){
    if(year() != 1970){
        if (AutoCirculation==1) {
            Blynk.syncVirtual(V8); // sync timeinput widget
        }
    }
}

BLYNK_WRITE(V2)  // All days selected
{
  if (param.asInt()==1) {
    Blynk.virtualWrite(V1, "clr");
    Blynk.virtualWrite(V1, "\n");
    Blynk.virtualWrite(V1,"\n Auto Circulation is Active");
    Blynk.virtualWrite(V1,"\n Please Wait...");
    Blynk.virtualWrite(V1, "\n");

    led1.on();
    led2.off();
    led3.off();
    led4.off();
    pinCheckStatus = 0;
    pinPhMode = 0;
    pinFanMode = 0;
    AutoCirculation=1;

    Blynk.virtualWrite(V4, 0);
    Blynk.virtualWrite(V24, 0);
    Blynk.virtualWrite(V20, 0);
    Blynk.virtualWrite(V26, 0);
    Blynk.virtualWrite(V30, 0);
    Blynk.virtualWrite(V31, 0);
  } else {
    AutoCirculation = 0;
    Blynk.virtualWrite(V1, "clr");
    Blynk.virtualWrite(V1, "\n");
    Blynk.virtualWrite(V1,"\n All Day Status : INACTIVE ");
    Blynk.virtualWrite(V1,"\n >> Press ON/OFF button if required
<<");
    Blynk.virtualWrite(V1, "\n");
    Blynk.virtualWrite(V1, "\n");
    Blynk.virtualWrite(V1, "\n");
   digitalWrite(RelayCirculation, LOW); // set LED OFF
   Blynk.virtualWrite(V4, 0);   //Turn OFF Button Widget
    led1.off();
  }
}
```

```
BLYNK_WRITE(V8)
{
  if (AutoCirculation==1) {
    sprintf(Date, "%02d/%02d/%04d",  day(), month(), year());
    sprintf(Time, "%02d:%02d:%02d", hour(), minute(), second());

    TimeInputParam t(param);

    Blynk.virtualWrite(V1,"clr");
    Blynk.virtualWrite(V1,"\n Date : ");
    Blynk.virtualWrite(V1,Date);
    Blynk.virtualWrite(V1,"\n All Days Checked schedule at: ");
    Blynk.virtualWrite(V1,Time);
    Blynk.virtualWrite(V1,"\n ALL DAYS ACTIVE today");

    if (t.hasStartTime()) { // Process start time
        Blynk.virtualWrite(V1,String("\n Start: ") +
t.getStartHour() + ":" + t.getStartMinute());
    }

    if (t.hasStopTime()) { // Process stop time
        Blynk.virtualWrite(V1,String("\n Stop : ") +
t.getStopHour() + ":" + t.getStopMinute());
    }

    Blynk.virtualWrite(V1,String("\n Time zone: ") + t.getTZ());
// Timezone is already added to start/stop time

    for (int i = 1; i <= 7; i++) {
// Process weekdays (1. Mon, 2. Tue, 3. Wed, ...)
          Blynk.virtualWrite(V1,String("\n Day ") + i + " is
selected");
    }

    nowseconds = (long(hour() * 3600L) + long(minute() * 60L) +
long(second()));
    Serial.println("NowSecond : ");  // used for debugging
    Serial.println(nowseconds);  // used for debugging

    startsecondswd = (long(t.getStartHour() * 3600L)) +
(long(t.getStartMinute() * 60L));
    Serial.println("StartSecond : ");  // used for debugging
    Serial.println(startsecondswd);  // used for debugging

    if(nowseconds >= startsecondswd){
        Blynk.virtualWrite(V1,"\n Penyiraman Aktif Pada Pukult :
");
        Blynk.virtualWrite(V1,String(" ") + t.getStartHour() + ":"
+ t.getStartMinute());
//        if(nowseconds <= startsecondswd + 90){
// 90s on 60s timer ensures 1 trigger command is sent
//            digitalWrite(RelayCirculation, HIGH);
// set LED ON
//            Blynk.virtualWrite(V4, 1);
//        }
    }
    else{
        Blynk.virtualWrite(V1, "\n");
```

```
        Blynk.virtualWrite(V1,"\n Penyiraman Belum Aktif Hari
ini"); //oke
    }

    stopsecondswd = (long(t.getStopHour() * 3600L)) +
(long(t.getStopMinute() * 60L) );
    Serial.println("StopSecond : ");  // used for debugging
    Serial.println(stopsecondswd);  // used for debugging

    if(nowseconds >= stopsecondswd){
        digitalWrite(RelayCirculation, LOW);
        Blynk.virtualWrite(V4, 0);
        Blynk.virtualWrite(V1,"\n Penyiraman Berhenti Pada Pukul
: ");
        Blynk.virtualWrite(V1,String(" ") + t.getStopHour() + ":"
+ t.getStopMinute());

        if(nowseconds <= stopsecondswd + 90){   // 90s on 60s
timer ensures 1 trigger command is sent
            digitalWrite(RelayCirculation, LOW);
            Blynk.virtualWrite(V4, 0);
        }
    }

    else{
        if(nowseconds >= startsecondswd){
            Blynk.virtualWrite(V1, "\n");
            Blynk.virtualWrite(V1,"\n Penyiraman Sedang
Berlangsung : "); // oke
            digitalWrite(RelayCirculation, HIGH); // set LED ON
TEST!!!!!
            Blynk.virtualWrite(V4, 1);
        }
    }
  }
}


void reconnectBlynk() {
  if (!Blynk.connected()) {
    if(Blynk.connect()) {
     BLYNK_LOG("Reconnected");
    } else {
      BLYNK_LOG("Not reconnected");
    }
  }
}

void loop()
{
  if (Blynk.connected()) {
    Blynk.run();
  }
  timer.run();
}
```