

DAFTAR PUSTAKA

Aris, Munandar. 2012. "Liquid Crystal Display 16x2 (LCD)". <http://www.leselektronika.com/2012/06/liquid-crystal-display-lcd-16-x-2.html>. Pada 19 Juli 2021, pukul 16.09 WIB.

Bahari, Gafur. H 2017. *Internet of Things untuk Pemantauan dan Pengendalian pada Sistem Hidroponik Nutrient Film Technique Sayuran Berbasis Wireless Sensor Network*".

Budiharto, Widodo dan Rizal, Gamel. 2007. *Belajar sendiri 12 Proyek Mikrokontroler Untuk Pemula*. Jakarta : PT. Elex Media Komputindo.

Hariri, Rafiq. 2019. *Perancangan Aplikasi Blynk Untuk Monitoring Dan Kendali Penyiraman Tanaman*. Jurnal Elektrikal Vol. 6 No. 1 (2019): Vol. 6, Juni 2019.

Ika, Kustanti. 2014. *Pengendalian Kadar Keasaman (pH) Pada Sistem Hidroponik Stroberi Menggunakan Kontroler PID Berbasis Arduino Uno*. Jurnal Ika Kustanti, 2014.

Istiqomah, Siti. 2006. *Menanam Hidroponik*. Jakarta: Azka Press.

Samadi, B. 2014. *Rahasia Budidaya Selada Secara Organik dan Anorganik*. Pustaka Mina, Jakarta.

Setyaningrum, H. dan C. Saporinto. 2011. *Panen Sayur Secara Rutin di Lahan Sempit*. Penebar Swadaya. Jakarta.

Siswadi, Teguh W. 2015. 'Pengaruh macam media terhadap pertumbuhan dan hasil selada (*Lactuca sativa* L.) hidroponik. Jurnal Agronomika 9(3), 257- 264.

Suhardiyanto, H. 2011. *Teknologi Hidroponik Untuk Budidaya Tanaman*. Fakultas Teknologi Pertanian, Bogor : IPB.

Wirman, Rahmi. 2019. *Kajian Tingkat Akurasi Sensor Pada Rancang Bangun Alat Ukur Total Dissolved Solid (TDS) Dan Tingkat Kekeruhan Air*. Jurnal Fisika, Universitas Negeri Semarang.

LAMPIRAN

Program Hardware di Software ArduinoIDE

```
#include <SoftwareSerial.h>

#include <EEPROM.h>

#include <OneWire.h>

#include <DallasTemperature.h>

#include "GravityTDS.h"

#define BLYNK_PRINT Serial

#define TINY_GSM_MODEM_SIM800

#include <TinyGsmClient.h>

#include <BlynkSimpleSIM800.h>

#define PIN_TEMPERATURE 8

#define PIN_TDS A0

#define PIN_PH A3

#define PIN_POMPA 9

char auth[] = "CFMU3EybxSHKF1QE--xdPiCjtwSOjgr_";

char apn[] = "internet";

char user[] = "";

char pass[] = "";

OneWire oneWire(PIN_TEMPERATURE);

GravityTDS gravityTds;
```

```
DallasTemperature sensors(&oneWire);

#include <SoftwareSerial.h>

SoftwareSerial SerialAT(52, 50);

TinyGsm modem(SerialAT);

void init_sensor() {

    pinMode(PIN_TEMPERATURE,INPUT);

    pinMode(PIN_PH,INPUT);

    pinMode(PIN_TDS,INPUT);

    pinMode(PIN_POMPA,OUTPUT);

    digitalWrite(PIN_POMPA,LOW);

    gravityTds.setPin(PIN_TDS);

    gravityTds.setAref(5.0);

    gravityTds.setAdcRange(1024);

    gravityTds.begin();

    sensors.begin();

}
```


0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00,

0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00,

0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00,

0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x80, 0x00, 0xc0, 0x60, 0x60, 0x30,
0x18, 0x84, 0x88,

0xc6, 0x43, 0x42, 0x61, 0x00, 0x51, 0x63, 0x44, 0xc6, 0xcc, 0x10, 0x08, 0x70,
0x60, 0x80, 0xc0,

0x80, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0xe0, 0xe0, 0xe0,

0xe0, 0xe0, 0xe0, 0x00, 0x00, 0xe0, 0xe0, 0xe0, 0xe0, 0xe0, 0x80, 0x00, 0x00,
0xe0, 0xe0, 0xe0,

0xe0, 0x00, 0x00, 0x00, 0xc0, 0xc0, 0xe0, 0xe0, 0xe0, 0xe0, 0xe0, 0xc0, 0x00,
0xe0, 0xe0, 0xe0,

0xe0, 0xe0, 0xe0, 0xe0, 0xe0, 0xe0, 0x00, 0xe0, 0xe0, 0xe0, 0xe0, 0xe0, 0x00,
0x00, 0xe0, 0xe0,

0xe0, 0xe0, 0xe0, 0xe0, 0xe0, 0xe0, 0xc0, 0x80, 0x00, 0x00, 0xe0, 0xe0, 0xe0,
0xe0, 0xe0, 0xe0,

0xe0, 0xe0, 0x00, 0x00, 0xe0, 0xe0, 0xe0, 0xe0, 0xe0, 0xe0, 0xe0, 0xc0, 0xc0,
0x00, 0x00, 0x00,

0x00, 0x30, 0xb8, 0x8c, 0x08, 0x06, 0x03, 0x00, 0x01, 0x00, 0xf0, 0x38, 0x98,
0x53, 0xe0, 0xf0,
0xfb, 0xfc, 0xfc, 0x3c, 0xcc, 0x3d, 0xfd, 0xfc, 0xfb, 0xf9, 0xcb, 0xe1, 0x18,
0x98, 0xe0, 0x00,
0x01, 0x02, 0x01, 0x06, 0x0c, 0xc4, 0xc8, 0x10, 0x00, 0x00, 0x00, 0x00, 0x00,
0xff, 0xff, 0xff,
0xff, 0xff, 0xff, 0x00, 0x00, 0xff, 0xff, 0xff, 0xff, 0xef, 0x7f, 0xfe, 0xf8, 0xff,
0xff, 0xff,
0xff, 0x00, 0x00, 0x0f, 0xbf, 0x3f, 0x3f, 0x7d, 0xf9, 0xf9, 0xfb, 0xf0, 0x00,
0x03, 0x03, 0xff,
0xff, 0xff, 0xff, 0xff, 0x03, 0x03, 0x00, 0xff, 0xff, 0xff, 0xff, 0xff, 0x00, 0x00,
0xff, 0xff,
0xff, 0xff, 0xf1, 0xf1, 0xff, 0xff, 0x7f, 0x3f, 0x00, 0x00, 0xff, 0xff, 0xff, 0xff,
0x79, 0x79,
0x79, 0x79, 0x00, 0x00, 0xff, 0xff, 0xff, 0xff, 0xf1, 0xf1, 0xff, 0xff, 0x3f, 0x1f,
0x00, 0x00,
0x00, 0x00, 0x01, 0x0d, 0x6c, 0x20, 0x40, 0x00, 0x00, 0x00, 0x1f, 0xf0, 0xec,
0x40, 0x9f, 0xff,
0x7f, 0x67, 0x11, 0x1b, 0xb4, 0xa0, 0x7b, 0xbf, 0x3f, 0xff, 0xaf, 0x9b, 0x60,
0x67, 0x03, 0x00,
0x00, 0x00, 0xc0, 0xf0, 0x36, 0x06, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x0f, 0x0f, 0x0f,

0x0f, 0x0f, 0x0f, 0x00, 0x00, 0x0f, 0x0f, 0x0f, 0x0f, 0x0f, 0x80, 0x81, 0x87,
0x8f, 0x0f, 0x0f,
0x8f, 0x80, 0x80, 0x8f, 0x0f, 0x0f, 0x9f, 0x9f, 0x8f, 0x8f, 0x07, 0x03, 0x80,
0x00, 0x80, 0x0f,
0x0f, 0x0f, 0x8f, 0x8f, 0x80, 0x00, 0x00, 0x8f, 0x8f, 0x8f, 0x8f, 0x0f, 0x00,
0x80, 0x8f, 0x8f,
0x0f, 0x0f, 0x00, 0x80, 0x80, 0x80, 0x00, 0x00, 0x00, 0x80, 0x8f, 0x8f, 0x0f,
0x0f, 0x0f, 0x8f,
0x8f, 0x0f, 0x00, 0x00, 0x0f, 0x0f, 0x0f, 0x0f, 0x00, 0x07, 0x0f, 0x0f, 0x0f,
0x0e, 0x08, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x03, 0x1b, 0xd8, 0xc0, 0x00, 0x00, 0x00, 0x01,
0x33, 0x32, 0x30,
0x33, 0x22, 0x01, 0x30, 0x00, 0x01, 0x01, 0x00, 0x23, 0x25, 0x34, 0x33, 0x00,
0x00, 0x00, 0x00,
0x60, 0x6c, 0x0d, 0x01, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x07, 0x07,
0x00, 0x00, 0x00,
0x07, 0x07, 0x07, 0x07, 0x00, 0x00, 0x07, 0x07, 0x06, 0x06, 0x00, 0x00, 0x01,
0x07, 0x01, 0x00,
0x00, 0x06, 0x07, 0x03, 0x07, 0x00, 0x00, 0x07, 0x07, 0x07, 0x06, 0x00, 0x00,
0x07, 0x07, 0x07,


```
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00
```

```
};
```

```
void drawWelcomeScreen(byte load = 1) {
    oled.clear();
    for (int i = 0; i < 8; i++) {
        oled.setCursor(0, i);
        for (int t = 0; t < 128; t++) {
            if (load){
                oled.ssd1306WriteRam(img[t + (128 * i)]);
            }
        }
    }
}
```

```
    }  
  }  
  if (load) {  
    oled.setCursor(13, 7);  
    oled.ssd1306WriteRam(0b11111111);  
    for (int t = 0; t < 100; t++) {  
      oled.ssd1306WriteRam(0b10000001);  
    }  
    oled.ssd1306WriteRam(0b11111111);  
    oled.setCursor(14, 7);  
    for (int t = 0; t < 100; t++) {  
      oled.ssd1306WriteRam(0b11111111);  
      delay(10);  
    }  
    //oled.clear();  
  }  
}
```

Program IoT(Internet Of Things) di Software ArduinoIDE

```
#include    "hardware.h"

#include    "display.h"

float temperature = 30, tdsValue = 0;

float calibration_value = 17.8;

int phval = 0;

float ph_nilai;

unsigned long int avgval;

int buffer_arr[10], temp;

BLYNK_CONNECTED() {

    Blynk.syncVirtual(V0);

    Blynk.syncVirtual(V1);

    Blynk.syncVirtual(V2);

}

void setup() {

    Serial.begin(9600);

    delay(10);

    init_display();

    drawWelcomeScreen();

    SerialAT.begin(19200);

    delay(1000);
```

```

//modem.restart();

Blynk.begin(auth, modem, apn, user, pass, "blynk-cloud.com", 8080);

init_sensor();

//init_display();

//drawWelcomeScreen();

oled.clear();

}

void loop() {

  Blynk.run();

  read_sensor();

  oled.set1X();

  oled.setCursor(23, 0); oled.print("IoT HIDROPONIK");

  oled.setCursor(0, 2); oled.print("Temp"); oled.setCursor(30, 2); oled.print(":");

oled.setCursor(37, 2); oled.print("      "); oled.setCursor(37, 2);

oled.print(sensors.getTempCByIndex(0)); oled.print(" C");

  oled.setCursor(0, 3); oled.print("pH"); oled.setCursor(30, 3); oled.print(":");

oled.setCursor(37, 3); oled.print("      "); oled.setCursor(37, 3);

oled.print(ph_nilai, 1);

  oled.setCursor(0, 4); oled.print("TDS"); oled.setCursor(30, 4); oled.print(":");

oled.setCursor(37, 4); oled.print("      "); oled.setCursor(37, 4);

oled.print(tdsValue, 0); oled.print(" ppm");

  oled.setCursor(0, 5); oled.print("Pompa"); oled.setCursor(30, 5); oled.print(":");

oled.setCursor(37, 5); oled.print("      ");

```

```
if (ph_nilai <= 5 || ph_nilai >= 8) {  
  
  //if (tdsValue > 100) {  
  
    oled.setCursor(37, 5); oled.print("ON");  
  
    digitalWrite(PIN_POMPA, HIGH);  
  
    WidgetLED led1(V3);  
  
    led1.on();  
  
  } else {  
  
    oled.setCursor(37, 5); oled.print("OFF");  
  
    digitalWrite(PIN_POMPA, LOW);  
  
    WidgetLED led1(V3);  
  
    led1.off();  
  
  }  
  
  Blynk.virtualWrite(V0, sensors.getTempCByIndex(0));  
  
  Blynk.virtualWrite(V1, ph_nilai);  
  
  Blynk.virtualWrite(V2, tdsValue);  
  
  delay(100);  
  
}  
  
void read_sensor() {  
  
  //temperature//  
  
  sensors.requestTemperatures();  
  
  
  //ph//  
  
  for (int i = 0; i < 10; i++)
```

```
{  
    buffer_arr[i] = analogRead(PIN_PH);  
    delay(30);  
}  
for (int i = 0; i < 9; i++)  
{  
    for (int j = i + 1; j < 10; j++)  
    {  
        if (buffer_arr[i] > buffer_arr[j])  
        {  
            temp = buffer_arr[i];  
            buffer_arr[i] = buffer_arr[j];  
            buffer_arr[j] = temp;  
        }  
    }  
}  
avgval = 0;  
for (int i = 2; i < 8; i++)  
    avgval += buffer_arr[i];  
float volt = (float)avgval * 5.0 / 1024 / 6;  
ph_nilai = -5.70 * volt + calibration_value;  
Serial.print("ADC pH: ");  
Serial.println(avgval);
```

```
Serial.print("V pH: ");  
  
Serial.println(volt);  
  
//tds//  
  
int adcTds = analogRead(PIN_TDS);  
  
float vTds = adcTds * (5.0 / 1024);  
  
Serial.print("ADC TDS: ");  
  
Serial.println(adcTds);  
  
Serial.print("V TDS: ");  
  
Serial.println(vTds);  
  
Serial.println(" ");  
  
gravityTds.setTemperature(temperature);  
  
gravityTds.update();  
  
tdsValue = gravityTds.getTdsValue();  
  
delay(500);  
  
}
```


HASIL UJI 2 SAMPLE AIR

NO	SAMPEL	Alat Pabrikan	Alat Rancangan	Selisih	Akurasi %	Error%
1	Air Sumur/Keran	27,7	27,06	0,64	97,68	0,23
2	Air Sungai	27,5	26,19	1,31	95,23	0,47
3	Air Sumur/Keran	27,7	26,94	0,76	97,25	0,27
4	Air Sungai	27,9	26,12	1,78	93,62	0,63
5	Air Sumur/Keran	27,8	26,87	0,93	96,65	0,33
6	Air Sungai	27,6	26,25	1,35	95,10	0,48
7	Air Sumur/Keran	27,7	26,81	0,89	96,78	0,32
8	Air Sungai	27,5	26,31	1,19	95,67	0,43
9	Air Sumur/ Keran	27,7	26,81	0,89	96,78	0,32
10	Air Sungai	27,5	26,44	1,06	96,14	0,38
Rata- Rata				1,08	96,09	0,38

NO	SAMPEL	Alat Pabrikan	Alat Rancangan	Selisih	Akurasi %	Error %
1	Air Sumur/Keran	7,29	7,0	0,29	96,02	3,97
2	Air Sungai	7,96	6,35	1,66	79,14	2,08
3	Air Sumur/Keran	7,09	6,92	0,19	97,32	2,67
4	Air Sungai	7,82	6,67	1,22	96,34	1,56
5	Air Sumur/Keran	7,11	7,38	0,19	95,67	2,67
6	Air Sungai	7,91	6,81	1,11	80,56	1,40
7	Air Sumur/Keran	7,15	6,95	0,25	96,45	3,49
8	Air Sungai	7,81	6,84	1,01	95,60	1,29
9	Air Sumur/Keran	7,15	6,82	0,35	94,35	4,89
10	Air Sungai	7,85	6,97	0,95	98,14	1,21
Rata- Rata				0,72	92,95%	2,52%

NO	SAMPEL	Alat Pabrikan	Alat Rancangan	Selisih	Akurasi %	Error %
1	Air Sumur	160	213	53	96,55	0,33
2	Air Sungai	160	122	40	95,14	0,25
3	Air Sumur	208	152	56	97,44	0,26
4	Air Sungai	166	120	46	98,58	0,27
5	Air Sumur	209	165	44	99,63	0,21
6	Air Sungai	164	122	42	90,47	0,25
7	Air Sumur	210	163	47	96,86	0,22
8	Air Sungai	159	122	37	94,32	0,23
9	Air Sumur	207	165	42	95,65	0,20
10	Air Sungai	161	132	29	97,32	0,18
Rata- Rata				43,6	96,19%	0,24%

GAMBAR KEGIATAN

A. Gambar Kegiatan Perakitan

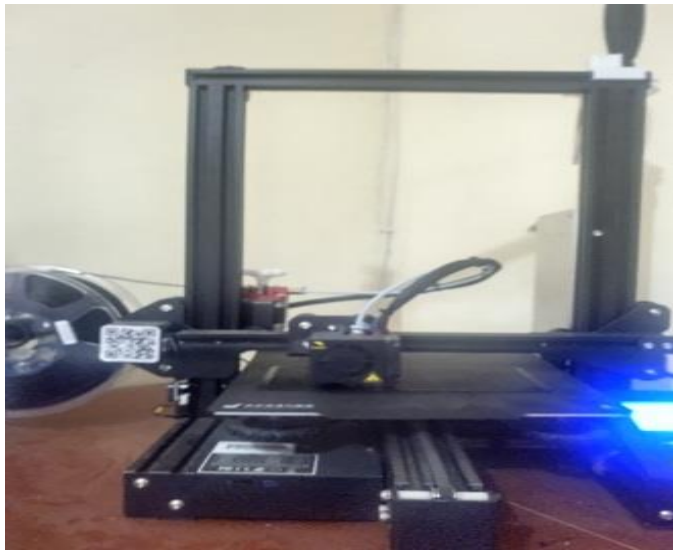
1. Tempat Perakitan Alat



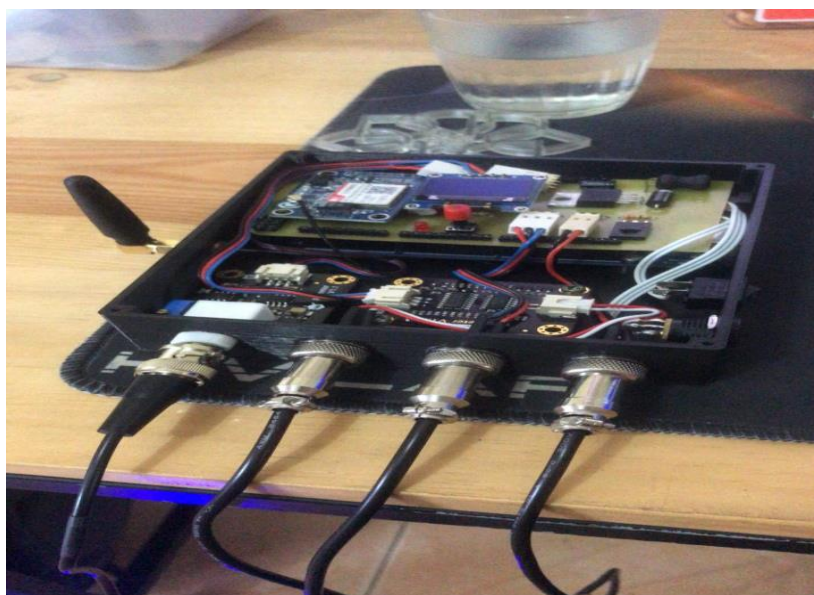
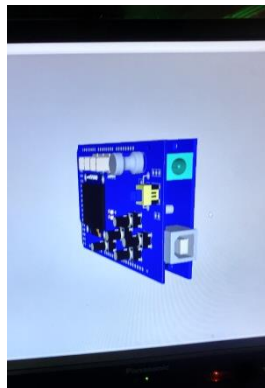
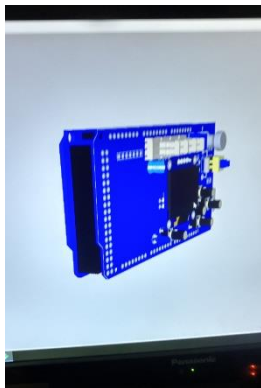
2. Perakitan Papan PCB(*Printed Circuit Board*)



3. Pembuatan Casing Alat



4. Perakitan Komponen Alat



5. Prototipe Alat Kualitas Air



B. Gambar Kegiatan Pengujian

1. Pengujian Sensor Suhu DS18B20



2. Pengujian Sensor pH DFRobot



3. Pengujian Sensor TDS(*Total Dissolved Solid*)



4. Pengujian Sampel Air Sungai



5. Pengujian sampel air hidroponik dengan Alat Rancangan



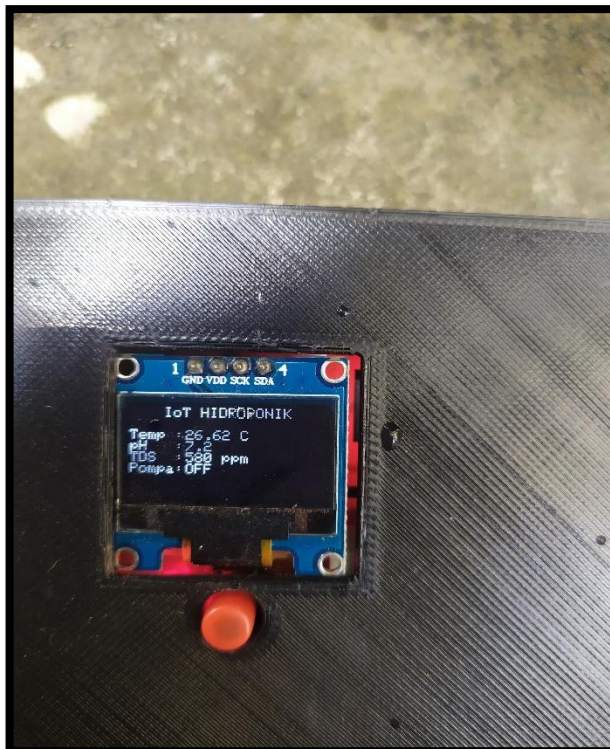
6. Pengujian sampel air hidroponik dengan Alat Pabrikan



7. Instalasi Hidroponik Yang digunakan



8. Hasil Pengujian Pada Alat Rancangan



9. Hasil Pengujian Sampel Air Pada Aplikasi *Blynk*



DATA SHEET SENSOR

Sensor TDS (*Total Dissolved Solid*)

Papan Pemancar Sinyal

Tegangan Masukan: 3.3 ~ 5.5V

Tegangan Keluaran: 0 ~ 2.3V

Bekerja Saat Ini: 3 ~ 6mA

Rentang Pengukuran TDS: 0 ~ 1000ppm

Akurasi Pengukuran TDS: $\pm 10\%$ F.S. (25)

Ukuran Modul: 42 * 32mm

Antarmuka Modul: PH2.0-3P

Antarmuka Elektroda: XH2.54-2P

pemeriksaan TDS

Jumlah Jarum: 2

Panjang Total: 83cm

Antarmuka Koneksi: XH2.54-2P

Warna: Hitam

Lainnya: Probe Tahan Air

Sensor pH DF Robot

Spesifikasi

Daya Modul: 5.00V

Ukuran Modul: 43mm \times 32mm

Rentang Pengukuran: 0-14PH

Mengukur Suhu: 0-60

Akurasi : ± 0.1 pH (25)

Waktu Respons: 1 menit

Sensor pH dengan Konektor BNC

Antarmuka PH2.0 (tambalan 3 kaki)

Dapatkan Potensiometer Penyesuaian

LED Indikator Daya

Panjang Kabel dari sensor ke konektor BNC: 660mm

Sensor Suhu DS18B20 *Waterproof*

Manfaat dan Fitur

Antarmuka 1-Kawat® Unik Hanya Membutuhkan Satu Port

Pin untuk Komunikasi Kurangi Jumlah Komponen dengan Terintegrasi Sensor Suhu dan EEPROM

Mengukur Suhu dari -55 °C hingga +125°C

(-67°F hingga +257°F)

$\pm 0,5^\circ\text{C}$ Akurasi dari -10°C hingga +85°C

Resolusi yang Dapat Diprogram dari 9 Bit hingga 12 Bit

- Tidak Diperlukan Komponen Eksternal

Mode Daya Parasit Hanya Membutuhkan 2 Pin untuk Operasi (DQ dan GND)

Menyederhanakan Sensor Suhu Terdistribusi Aplikasi dengan Kemampuan Multidrop

- Setiap Perangkat Memiliki Kode Serial 64-Bit Unik

Disimpan di ROM On-Board

- Pengaturan Alarm Nonvolatile (NV) yang Dapat Ditentukan Pengguna Fleksibel dengan Perintah Pencarian Alarm Mengidentifikasi Perangkat dengan Suhu Di Luar Batas Terprogram

- Tersedia dalam 8-Pin SO (150 mils), SOP, dan . 8-Pin

Paket 3-Pin TO-92