

**MODUL PEMBELAJARAN PEMROGRAMAN IOT**  
**DOKUMENTASI SISTEM IOT PADA GREENHOUSE**



**Disusun oleh :**

Siti Marfugah Dwi Lestari

20210140038

**PROGRAM STUDI TEKNOLOGI INFORMASI**  
**FAKULTAS TEKNIK**  
**UNIVERSITAS MUHAMMADIYAH YOGYAKARTA**  
**2025**

# PENGANTAR

Dokumentasi ini disusun sebagai bagian dari tugas akhir dengan tujuan utama mendukung pembelajaran pemrograman Internet of Things (IoT) di lingkungan greenhouse. Isinya merupakan hasil pengalaman langsung selama proses pengembangan sistem IoT berbasis Raspberry Pi yang digunakan untuk memantau suhu, kelembaban, intensitas cahaya, dan pH tanah.

Dokumen ini sengaja dibuat agar bisa diikuti oleh mahasiswa lain yang ingin belajar atau mengembangkan sistem serupa. Penulis mencoba menyajikan langkah-langkahnya secara runtut dan mudah dipahami, mulai dari pengenalan alat, perakitan hardware, pemrograman, hingga koneksi jaringan dan pengiriman data.

Harapannya, dokumentasi ini bisa menjadi panduan praktis, bukan hanya sebagai pelengkap skripsi, tapi juga sebagai media belajar langsung bagi teman-teman yang tertarik dengan IoT, khususnya dalam konteks greenhouse.

Penulis menyadari masih banyak kekurangan dalam penyusunan dokumen ini, tapi diharapkan tetap bisa memberikan gambaran yang cukup jelas dan membantu.

# DAFTAR ISI

PENGANTAR .....	2
DAFTAR ISI .....	3
BAB I. PENDAHULUAN.....	4
1.1. Latar Belakang.....	4
1.2. Tujuan .....	4
1.3. Ruang Lingkup .....	5
1.4. Sasaran Pembaca.....	5
BAB II. PENGENALAN PERANGKAT DAN ARSITEKTUR SISTEM .....	6
2.1 Komponen Utama Sistem.....	6
2.2 Alat dan Bahan yang Digunakan .....	13
2.3 Skema Arsitektur Sistem .....	15
BAB III. INSTALASI DAN KONFIGURASI SISTEM .....	16
3.1 Instalasi dan Konfigurasi Raspberry Pi.....	16
3.2 Wiring dan Perakitan Hardware .....	16
3.3 Diagram Sistem Catu Daya.....	18
BAB IV. PEMROGRAMAN DAN KALIBRASI SENSOR.....	19
4.1 Struktur Program Python.....	19
4.2 Pengujian Pengiriman Data .....	26
BAB V. MONITORING JARAK JAUH.....	34

# **BAB I.**

## **PENDAHULUAN**

### **1.1. Latar Belakang**

Internet of Things (IoT) merupakan salah satu perkembangan teknologi yang semakin luas diterapkan dalam berbagai bidang, termasuk pertanian. Salah satu contohnya adalah pemantauan kondisi lingkungan secara otomatis pada greenhouse. Dengan bantuan berbagai sensor dan perangkat mikrokontroler seperti Raspberry Pi, petani atau peneliti dapat memantau suhu, kelembaban, cahaya, dan pH tanah secara real-time tanpa harus hadir langsung di Lokasi.

Namun di sisi lain, pemrograman IoT bukanlah hal yang mudah dipelajari tanpa praktik langsung. Mahasiswa sering mengalami kesulitan saat harus menghubungkan perangkat keras dan menulis program yang berfungsi secara utuh. Untuk itulah dokumentasi ini dibuat.

Dokumentasi ini disusun sebagai panduan teknis bagi mahasiswa dalam memahami dan membangun sistem IoT sederhana yang digunakan untuk memantau kondisi greenhouse. Selain menjadi lampiran dari skripsi, dokumen ini diharapkan dapat menjadi media pembelajaran yang aplikatif dan mudah dipraktikkan.

### **1.2. Tujuan**

Dokumentasi ini bertujuan untuk:

- Memahami konsep dasar IoT dan aplikasinya pada sistem greenhouse
- Mengidentifikasi komponen-komponen utama sistem IoT untuk greenhouse
- Merakit perangkat keras sistem IoT untuk pemantauan dan kontrol greenhouse
- Melakukan konfigurasi perangkat IoT dan menghubungkannya ke jaringan

### **1.3. Ruang Lingkup**

Dokumentasi ini berfokus pada sistem IoT yang terdiri dari Raspberry Pi, sensor suhu, dan kelembaban (DHT22), sensor cahaya (BH1750), sensor pH tanah (ph probe analytical surver electrode), serta sistem komunikasi menggunakan VPN (Tailscale) dan modem USB. Pembahasan mencakup instalasi, wiring, pemrograman, dan pengiriman data ke platform web ilmuku.id. visualisasi data di web tidak dibahas secara rinci karena sistem website sudah tersedia dan tidak dikembangkan oleh penulis.

### **1.4. Sasaran Pembaca**

Dokumentasi ini ditujukan untuk:

- Mahasiswa yang ingin belajar pemrograman dan perakitan sistem IoT dari dasar.
- Dosen atau pengajar yang ingin menggunakan sistem ini sebagai media ajar.
- Siapa saja yang tertarik dengan pengembangan sistem monitoring lingkungan berbasis IoT.

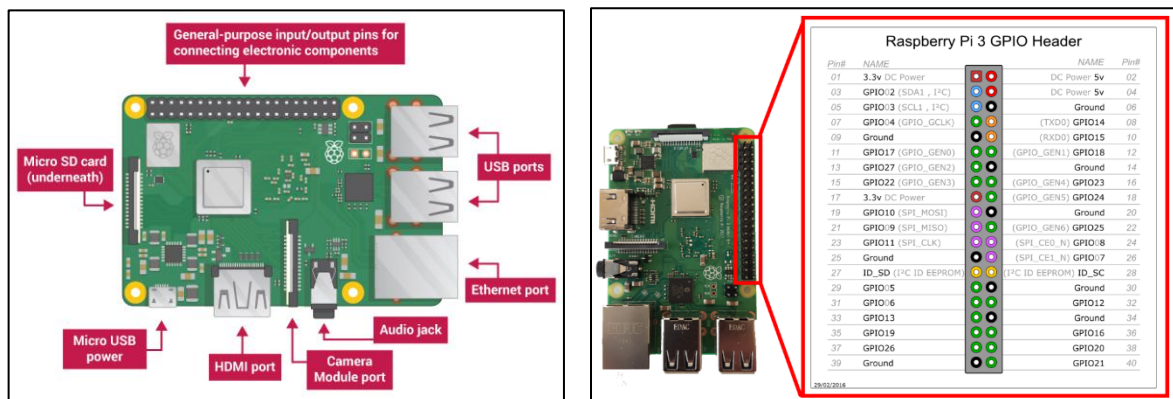
# BAB II.

## Pengenalan Perangkat dan Arsitektur Sistem

### 2.1 Komponen Utama Sistem

Sistem IoT dibangun dengan tujuan untuk memantau kondisi lingkungan dalam greenhouse secara otomatis dan dapat diakses jarak jauh. Komponen utama yang digunakan antara lain:

#### 1. Raspberry Pi 3 Model B+



#### Spesifikasi teknis :

- Prosesor: Broadcom BCM2837B0, Cortex-A53 (ARMv8) 64-bit SoC @ 1.4GHz
- Memori: 1GB LPDDR2 SDRAM
- Konektivitas: WiFi dual-band (2.4GHz dan 5GHz), Bluetooth 4.2, BLE
- GPIO: Header GPIO 40-pin
- Daya: Input DC 5V/2.5A

#### Peran dalam Sistem:

Raspberry Pi 3 Model B+ berperan sebagai otak utama dalam sistem. Perangkat ini menjalankan program Python untuk membaca data dari sensor, mengatur waktu pembacaan data, serta mengirimkan data ke data logger. Selain itu, Raspberry Pi juga

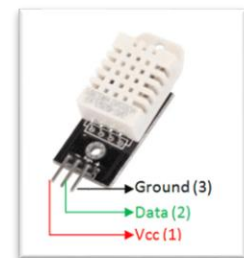
berfungsi sebagai titik akses jarak jauh melalui VPN dan sebagai pusat kendali sistem secara keseluruhan.

### **Keunggulan dalam Proyek IoT:**

- Sistem Operasi Berbasis Linux: Memudahkan pengembangan program dan instalasi berbagai pustaka.
- Banyak Pilihan Input/Output: Mendukung GPIO, I2C, SPI, dan UART untuk komunikasi dengan sensor.
- Konektivitas Terintegrasi: Sudah dilengkapi dengan WiFi dan Bluetooth untuk komunikasi nirkabel.
- Dukungan Komunitas: Tersedia dokumentasi dan pustaka yang melimpah dari komunitas global.
- Efisiensi Daya: Cocok digunakan pada sistem dengan sumber daya terbatas seperti baterai atau panel surya.

## **2. Sensor DHT22**

Berfungsi untuk mengukur suhu dan kelembaban udara. Sensor ini memiliki akurasi cukup baik untuk kebutuhan pemantauan lingkungan tanaman.



### **Spesifikasi Teknis:**

- Rentang Suhu:  $-40^{\circ}\text{C}$  hingga  $80^{\circ}\text{C}$  (akurasi  $\pm 0,5^{\circ}\text{C}$ )
- Rentang Kelembaban: 0–100% RH (akurasi  $\pm 2-5\%$ )
- Tegangan Operasi: 3,3V hingga 6V DC
- Output: Sinyal digital dengan protokol satu kabel (single-bus)
- Kecepatan Sampling: 0,5Hz (setiap 2 detik sekali)

### **Protokol Komunikasi:**

DHT22 menggunakan protokol komunikasi satu kabel (single-wire) yang bersifat khusus. Proses komunikasinya meliputi:

- Inisialisasi: Mikrokontroler (MCU) mengirim sinyal start dengan menurunkan tegangan (LOW) selama 1–10 milidetik.
- Respon Sensor: Sensor merespon dengan sinyal LOW selama 80 mikrodetik, lalu HIGH selama 80 mikrodetik.
- Pengiriman Data: Data dikirim dalam bentuk 40 bit, terdiri dari 16 bit kelembaban, 16 bit suhu, dan 8 bit checksum.

### 3. Sensor BH1750

Digunakan untuk mengukur intensitas cahaya di dalam greenhouse. Sensor ini berkomunikasi melalui protokol I2C dan menghasilkan data dalam satuan lux.



#### Spesifikasi Teknis:

- Rentang Pengukuran: 0 hingga 65.535 lux
- Resolusi: 0,5 lux (pada mode H-resolution)
- Tegangan Operasi: 2,4V hingga 3,6V
- Antarmuka Komunikasi: I2C (Inter-Integrated Circuit)
- Alamat I2C: 0x23 jika pin ADDR dalam kondisi LOW, atau 0x5C jika pin ADDR dalam kondisi HIGH

#### Mode Pengukuran:

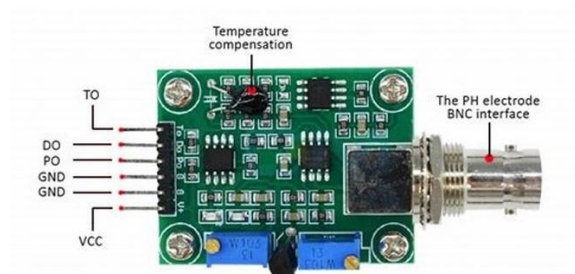
- Sensor BH1750 menyediakan beberapa mode pengukuran yang dapat disesuaikan dengan kebutuhan aplikasi. Mode tersebut antara lain:
- Continuous H-Resolution Mode: Menghasilkan data setiap 120 milidetik dengan resolusi 1 lux. Cocok untuk pengukuran berkelanjutan dengan ketelitian sedang.



- Continuous H-Resolution Mode2: Memiliki resolusi lebih tinggi yaitu 0,5 lux, dengan waktu pengukuran yang sama (120 ms). Mode ini sangat ideal untuk aplikasi yang membutuhkan akurasi tinggi.
- Continuous L-Resolution Mode: Memiliki resolusi 4 lux dengan waktu pengukuran lebih cepat, yaitu 16 ms. Mode ini cocok untuk sistem yang memprioritaskan kecepatan daripada presisi.
- One-Time Modes: Sensor melakukan satu kali pengukuran kemudian masuk ke mode tidur (power down). Mode ini efisien untuk sistem yang hemat daya seperti aplikasi berbasis baterai

#### 4. pH Probe + pH Module

Digunakan untuk mengukur Tingkat keasaman (pH) tanah. Karena output sensor ini berupa sinyal analog, maka diperlukan ADC tambahan.



##### **Komponen Utama:**

- Sistem pengukuran pH terdiri dari tiga bagian utama, yaitu:
- Elektrode pH: Sensor berbasis elektroda kaca yang dilengkapi referensi internal, berfungsi mendeteksi konsentrasi ion hidrogen ( $H^+$ ) dalam larutan.
- Modul pH: Bertugas memperkuat dan mengkondisikan sinyal analog dari elektroda agar dapat dibaca oleh mikrokontroler.
- Konektor BNC: Digunakan untuk menyambungkan elektroda ke modul dengan perlindungan terhadap gangguan sinyal (noise).

### Spesifikasi Teknis:

- Rentang Pengukuran: pH 0–14
- Akurasi:  $\pm 0,1$  pH (setelah proses kalibrasi)
- Kompensasi Suhu: Manual atau otomatis, tergantung jenis modul yang digunakan
- Output: Tegangan analog (umumnya antara 0 hingga 3,3V)
- Waktu Respon: Kurang dari 1 menit untuk mencapai 90% pembacaan stabil

### Proses Kalibrasi:

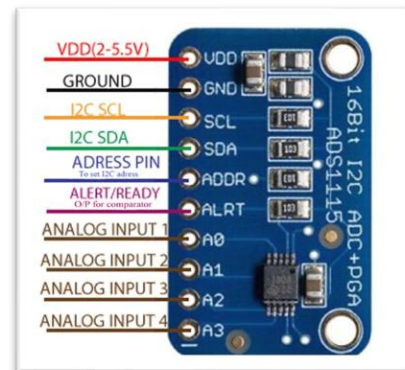
- Kalibrasi 3 Titik: Menggunakan larutan buffer pH 4.01, 6.86, dan 9.18
- Perhitungan Slope: Menentukan rasio mV per unit pH berdasarkan hasil pengukuran buffer
- Penyesuaian Offset: Menyesuaikan titik netral (pH 7.00) agar tegangan sesuai standar
- Kompensasi Suhu: Dilakukan untuk mengoreksi perubahan slope akibat perubahan suhu lingkungan

## 5. ADS1115 (Analog to Digital Converter)

Berfungsi untuk mengkonversi sinyal analog dari sensor pH menjadi data digital agar dapat dibaca oleh Raspberry Pi yang tidak memiliki input analog.

### Spesifikasi Teknis:

- Resolusi: 16-bit (65.536 level pengukuran)
- Jumlah Kanal: 4 kanal single-ended atau 2 kanal diferensial
- Kecepatan Sampling: 8 hingga 860 sampel per detik
- Penguatan yang Dapat Diprogram (Programmable Gain): Rentang tegangan penuh dari  $\pm 6,144V$  hingga  $\pm 0,256V$
- Antarmuka Komunikasi: I2C
- Alamat I2C: 0x48 hingga 0x4B (dapat disesuaikan melalui pin ADDR)



**Fungsi dalam Sistem:**

ADS1115 digunakan untuk membaca sinyal analog dari sensor yang tidak memiliki output digital, seperti modul pH analog atau sensor kelembaban tanah kapasitif. Karena Raspberry Pi tidak memiliki pin analog bawaan, ADS1115 menjadi komponen penting untuk konversi sinyal analog menjadi digital agar bisa diolah oleh sistem.

**Keunggulan ADS1115:**

- Akurasi Tinggi: Dengan resolusi 16-bit, ADS1115 mampu mendeteksi perubahan tegangan yang sangat kecil, cocok untuk sensor yang membutuhkan presisi.
- Konfigurasi Fleksibel: Dapat digunakan untuk pengukuran tegangan dari empat sumber berbeda secara individual (single-ended), atau dua pasang sumber tegangan (differential).
- Penguatan Terprogram: Nilai penguatan (gain) dapat diatur sesuai kebutuhan, memungkinkan pembacaan tegangan rendah dengan lebih akurat.
- Komunikasi Mudah: Berbasis I2C sehingga dapat langsung terhubung ke Raspberry Pi dengan hanya dua jalur (SCL dan SDA).
- Dukungan Multi-Alamat: Beberapa modul ADS1115 dapat digunakan sekaligus dengan mengatur alamat I2C yang berbeda.

## 6. Tailscale VPN

**Pengertian:**

Tailscale adalah layanan VPN berbasis protokol WireGuard yang memungkinkan perangkat saling terhubung melalui jaringan pribadi yang aman, tanpa perlu konfigurasi jaringan rumit seperti port forwarding atau IP publik. Tailscale menyederhanakan pembentukan koneksi peer-to-peer antar perangkat, meskipun berada di jaringan yang berbeda.

**Fungsi dalam Sistem:**

Dalam proyek IoT ini, Tailscale digunakan untuk memberikan akses jarak jauh ke Raspberry Pi yang berada di lokasi greenhouse. Dengan VPN ini, pengguna dapat:

- Mengakses terminal Raspberry Pi melalui SSH secara aman dari mana pun
- Memantau dan mengontrol sistem tanpa harus berada di lokasi fisik
- Melakukan debugging atau update kode langsung dari laptop atau PC
- Mengirim data ke server atau mengambil data log secara real-time

### **Cara Kerja Tailscale:**

- Setiap perangkat (laptop, Raspberry Pi, HP) yang terhubung ke Tailscale akan mendapatkan IP privat dari jaringan Tailscale.
- Setelah login dengan akun (misalnya akun Google), perangkat akan otomatis terhubung ke jaringan virtual yang sama.
- Semua koneksi antar perangkat dienkripsi dan bersifat end-to-end, sehingga aman meskipun menggunakan jaringan publik seperti WiFi umum.

### **Keunggulan Tailscale:**

- Instalasi Mudah: Hanya perlu instal aplikasi dan login, tidak perlu konfigurasi jaringan manual.
- Akses Aman: Semua komunikasi terenkripsi menggunakan standar WireGuard.
- Peer-to-Peer: Tidak bergantung pada server pusat untuk lalu lintas data, sehingga lebih cepat dan efisien.
- Multi-Platform: Mendukung Windows, Linux, macOS, Android, dan iOS.
- Firewall Friendly: Tidak perlu mengatur port forwarding di router, karena koneksi dibuat secara otomatis melalui NAT traversal.

### **Penerapan pada Greenhouse IoT:**

Dengan Tailscale, Raspberry Pi di greenhouse tetap bisa diakses dan dikendalikan dari luar meskipun hanya menggunakan koneksi internet biasa (misalnya modem seluler). Hal ini sangat penting karena sistem harus tetap bisa dipantau dan dikelola secara jarak jauh untuk memastikan fungsinya berjalan dengan baik.

## 2.2 Alat dan Bahan yang Digunakan

Pada pengembangan sistem IoT untuk monitoring greenhouse ini, digunakan sejumlah perangkat keras dan perangkat lunak sebagai alat bantu dan bahan utama. Semua komponen dipilih berdasarkan sistem, keterjangkauan, dan kemudahan dalam implementasi.

### A. Perangkat Keras (Hardware)

No	Nama Komponen	Fungsi
1.	Raspberry Pi 3 Model B+	Mikrokontroler utama, menjalankan program Python dan pengolahan data.
2.	MicroSD Card 8 GB + Card Reader	Penyimpanan sistem operasi dan program pada Raspberry Pi.
3.	Sensor DHT22	Mengukur suhu dan kelembaban udara
4.	Sensor BH1750	Mengukur intensitas cahaya dalam satuan lux
5.	Sensor pH probe + pH module	Mengukur keasaman tanah dengan output analog
6.	ADS1115 16-bit ADC	Konversi sinyal analog ke digital agar terbaca oleh Raspberry Pi
7.	Breadboard 400 pin	Media penyambungan sementara komponen elektronik oleh Raspberry Pi
8.	Kabel Jumper (male to male, female to male)	Menghubungkan komponen ke Raspberry Pi
9.	Heatsink Fan	Pendingin Raspberry Pi untuk menjaga suhu tetap stabil
10.	Kabel USB to MicroUSB	Kabel daya untuk Raspberry Pi
11.	Panel Surya 30W	Sumber energi terbarukan
12.	Baterai / Aki	Menyimpan daya dari panel surya
13.	Solar Charge Controller (SCC)	Mengatur pengisian baterai dari panel surya
14.	Modem USB + Kartu SIM	Koneksi Internet untuk mengirim data ke website

15.	Kotak/Case tahan air	Melindungi perangkat dari debu dan air di lapangan
16.	Tiang Penyangga Panel Surya	Menopang panel surya di area outdoor
17.	Mur, baut, dan kabel SCC	Perlengkapan pemasangan sistem daya

#### B. Alat Penunjang

No	Nama Alat	Fungsi
1.	Laptop	sebagai client untuk mengakses sistem IoT jarak jauh setelah konfigurasi VPN, serta untuk mengembangkan kode.
2.	Monitor, keyboard, mouse, kabel HDMI	Setup awal Raspberry Pi
3.	Solder dan timah solder	Menyambung pin pada komponen
4.	Toolset elektronik	Obeng, tang, pemotong kabel, bor kecil

#### C. Perangkat Lunak

No	Software	Fungsi
1.	Raspbian Pi OS (Raspbian)	Sistem operasi utama Raspberry Pi
2.	Python 3	Bahasa pemrograman utama untuk membaca dan kirim data
3.	Tailscale VPN	Mengakses Raspberry Pi dari jarak jauh secara aman
4.	Website ilmuku.id	Platform data logger untuk menyimpan dan menampilkan data

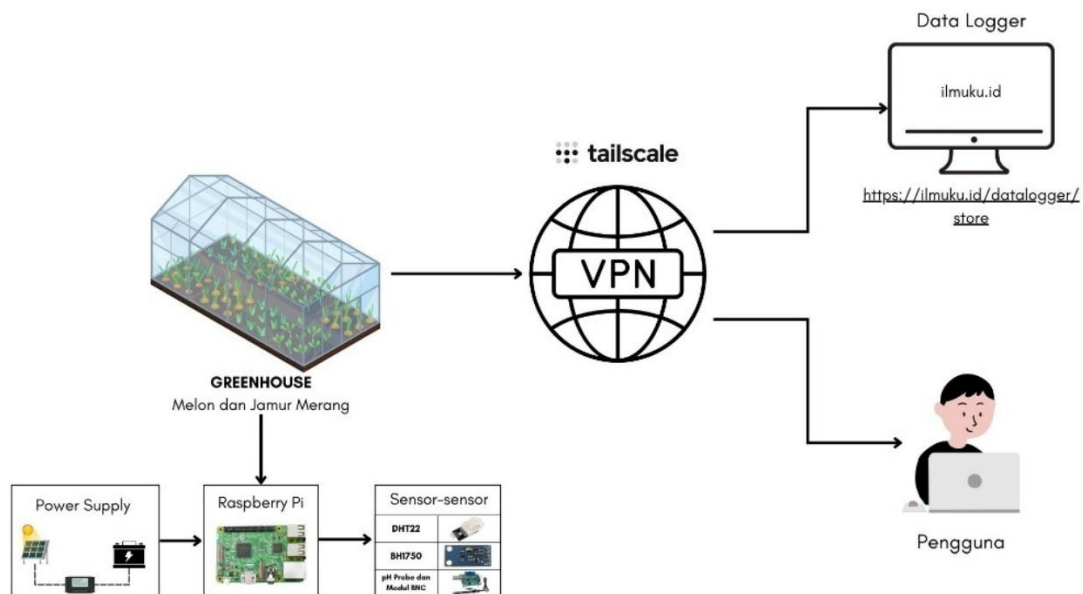
#### D. Bahan Kalibrasi

No	Bahan	Fungsi
1.	Larutan Buffer pH 4.01	Kalibrasi sensor titik rendah
2.	Larutan Buffer pH 6.86	Kalibrasi sensor titik tengah
3.	Larutan Buffer pH 9.18	Kalibrasi sensor titik tinggi



## 2.3 Skema Arsitektur Sistem

Sistem ini terdiri dari beberapa bagian yang saling terhubung dan bekerja secara berurutan. Berikut alur sistemnya:



1. Sensor (DHT22, BH1750, pH Probe) membaca data lingkungan
2. ADS1115 mengubah sinyal analog dari sensor pH menjadi data digital
3. Raspberry Pi mengumpulkan data dari semua sensor
4. Program Python di Raspberry Pi memproses data dan mengirimkannya ke website ilmuku.id
5. Raspberry Pi terhubung ke internet melalui modem USB
6. Data ditampilkan dalam bentuk tabel dan grafik pada web, yang dapat diakses dari jarak jauh oleh pengguna
7. Akses remote Raspberry Pi juga dapat dilakukan melalui VPN (Tailscale) menggunakan laptop client.

# BAB III.

## INSTALASI DAN KONFIGURASI SISTEM

### 3.1 Instalasi dan Konfigurasi Raspberry Pi

#### 1. Instalasi Sistem Operasi

- a. Hubungkan microSD card ke laptop menggunakan card reader.
- b. Kunjungi situs <https://www.raspberrypi.org/software> dan unduh Raspberry Pi Imager.
- c. Jalankan Raspberry Pi Imager
- d. Pilih “Raspberry Pi OS (32-bit)” pada opsi OS
- e. Pilih storage (microSD)
- f. Klik **WRITE**, tunggu proses selesai
- g. Setelah selesai, lepaskan microSD dari laptop

#### 2. Menyiapkan Raspberry Pi

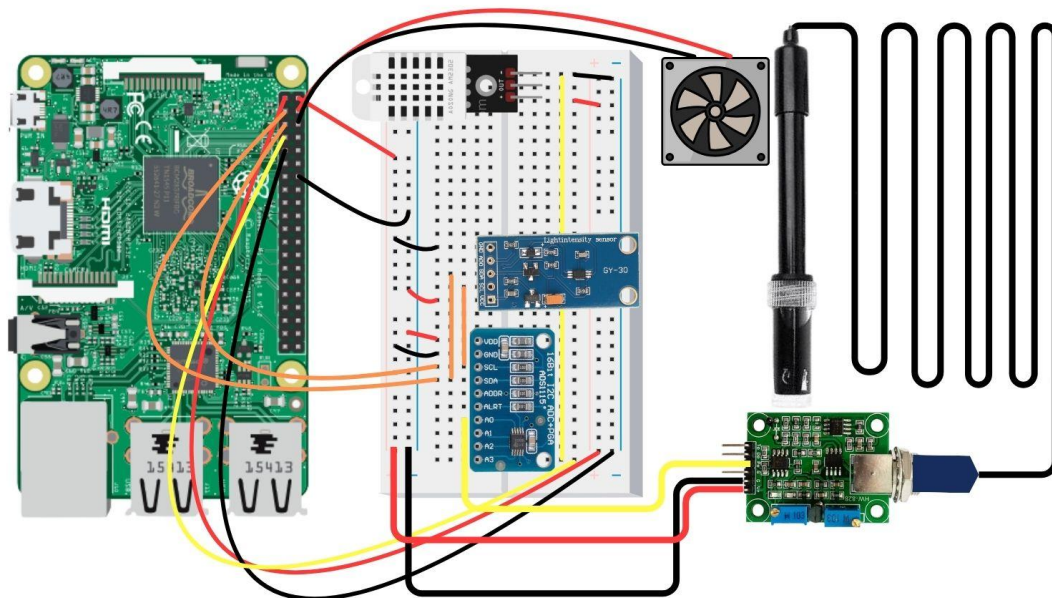
- a. Masukkan microSD ke slot Raspberry Pi
- b. Hubungkan Raspberry Pi ke monitor melalui kabel HDMI
- c. Sambungkan keyboard dan mouse
- d. Hubungkan kabel power untuk menyalakan Raspberry Pi
- e. Ikuti wizard awal : atur bahasa, zona waktu, password, dan sambungkan ke Wi-Fi
- f. Buka terminal dan jalankan update:

```
sudo apt update && sudo apt upgrade -y
```

### 3.2 Wiring dan Perakitan Hardware

Langkah berikutnya adalah menyusun koneksi antara Raspberry Pi dan sensor-sensor menggunakan breadboard dan kabel jumper. Semua sensor dipasang pada breadboard dan dihubungkan ke Raspberry Pi melalui GPIO (General Purpose Input Output).





**Koneksi DHT22 dengan Raspberry Pi:**

DHT22	Raspberry Pi
VCC (+)	3.3V (pin 1)
Data	GPIO 17 (pin 11)
GND (-)	GND (pin 9)

**Koneksi BH1750 dengan Raspberry Pi:**

BH1750	Raspberry Pi
VCC	5V (pin 2)
SCL	GPIO 3 (pin 5)
SDA	GPIO 2 (pin 3)
GND	GND (pin 14)

### Koneksi ADS1115 dengan Raspberry Pi:

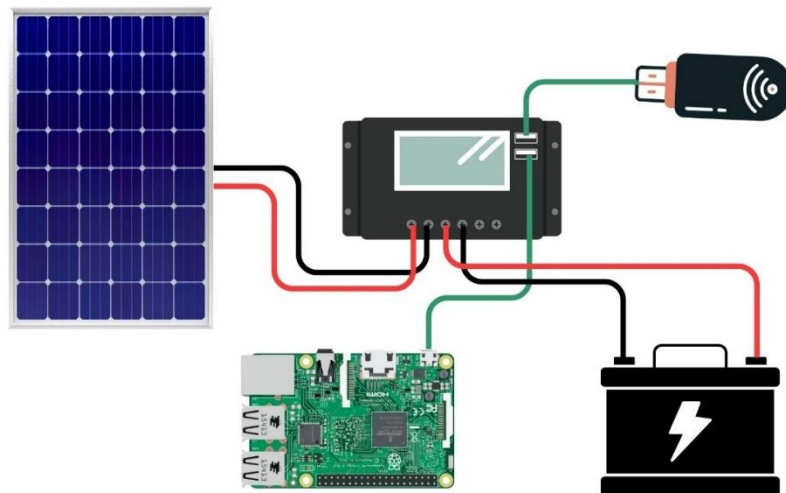
ADS1115	Raspberry Pi
VDD	5V (pin 2)
SCL	GPIO 3 (pin 5)
SDA	GPIO 2 (pin 3)
GND	GND (pin 14)

### Koneksi dengan pH Probe dengan ADS1115:

pH Probe	ADS1115
V+	VDD
G	GND
Po	A0

### 3.3 Diagram Sistem Catu Daya

Sistem mendapatkan daya dari panel surya dan aki yang terhubung ke solar charge controller (SCC). Dari SCC, output dihubungkan ke port USB sebagai sumber daya untuk Raspberry Pi dan Modem. Skema ini memungkinkan sistem berjalan mandiri tanpa Listrik PLN.



## **BAB IV.**

# **PEMROGRAMAN DAN KALIBRASI SENSOR**

Setelah proses instalasi dan perakitan perangkat keras selesai, tahap selanjutnya adalah mengembangkan perangkat lunak yang bertugas membaca data dari sensor, mengolahnya, dan mengirimkan ke server (platform web) untuk visualisasi. Selain itu, kalibrasi sensor dilakukan agar data yang dikumpulkan mempresentasikan kondisi lingkungan sebenarnya dengan akurasi yang dapat diandalkan, terutama untuk sensor analog seperti sensor pH.

### **4.1 Struktur Program Python**

#### **1. Inisiasi dan Import Library**

##### **DHT22**

```
sudo pip3 install adafruit-circuitpython-dht  
sudo apt-get install libgpod2
```

##### **BH1750**

```
sudo pip3 install adafruit-blinka  
sudo pip3 install adafruit-circuitpython-busdevice
```

##### **ADS1115 dan sensor pH**

```
sudo pip3 install adafruit-circuitpython-ads1x15
```

#### **2. Pembacaan Data Sensor**

Berikut ini adalah contoh kode Python untuk membaca data dari sensor-sensor yang digunakan:

## A. DHT22 untuk Suhu dan Kelembaban

```
GNU nano 7.2          suhu.py
import time
import board
import adafruit_dht

# Gunakan GPIO yang sesuai (contoh: GPIO4)
dhtDevice = adafruit_dht.DHT22(board.D4)

print("Mulai membaca suhu dan kelembaban dari DHT22...")
print("Tekan Ctrl+C untuk berhenti.")

try:
    while True:
        temperature = dhtDevice.temperature
        humidity = dhtDevice.humidity
        if humidity is not None and temperature is not None:
            print(f"Suhu: {temperature:.1f}°C Kelembaban: {humidity:.1f}%")
        else:
            print("Gagal membaca dari sensor, coba lagi...")
            time.sleep(2.0)
except KeyboardInterrupt:
    print("Dihentikan oleh pengguna")
except Exception as error:
    print("Terjadi kesalahan:", error)
finally:
    dhtDevice.exit()
```

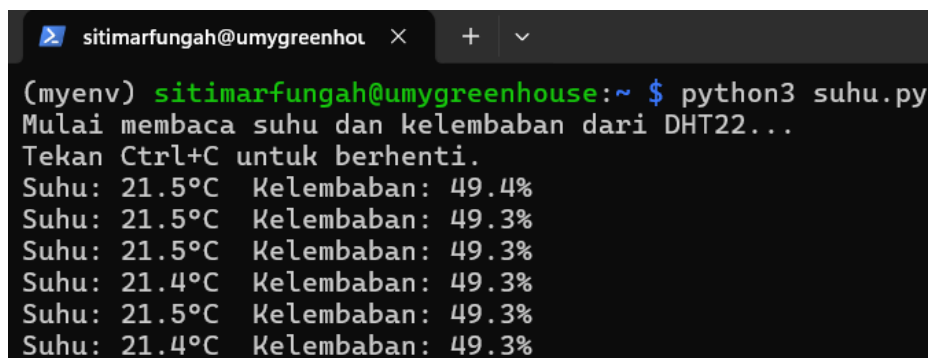
Code ini adalah program Python untuk membaca sensor suhu dan kelembaban DHT22 menggunakan mikrocontroller yang kompatibel dengan library Adafruit CircuitPython. Program dimulai dengan mengimpor library yang diperlukan yaitu `time` untuk delay, `board` untuk mengakses pin GPIO, dan `adafruit_dht` untuk berkomunikasi dengan sensor DHT22. Sensor dihubungkan ke pin D4 pada board dan diinisialisasi sebagai objek `dhtDevice`.

Program kemudian menampilkan pesan pembuka dan masuk ke dalam loop tak terbatas yang terus membaca data sensor. Di dalam loop, program mencoba membaca nilai temperatur dan kelembaban dari sensor, lalu memeriksa apakah kedua nilai tersebut valid (tidak `None`). Jika pembacaan berhasil, data akan ditampilkan dalam format yang mudah dibaca dengan satu angka desimal. Namun jika sensor gagal dibaca,

program akan menampilkan pesan error dan melanjutkan ke iterasi berikutnya setelah delay 2 detik.

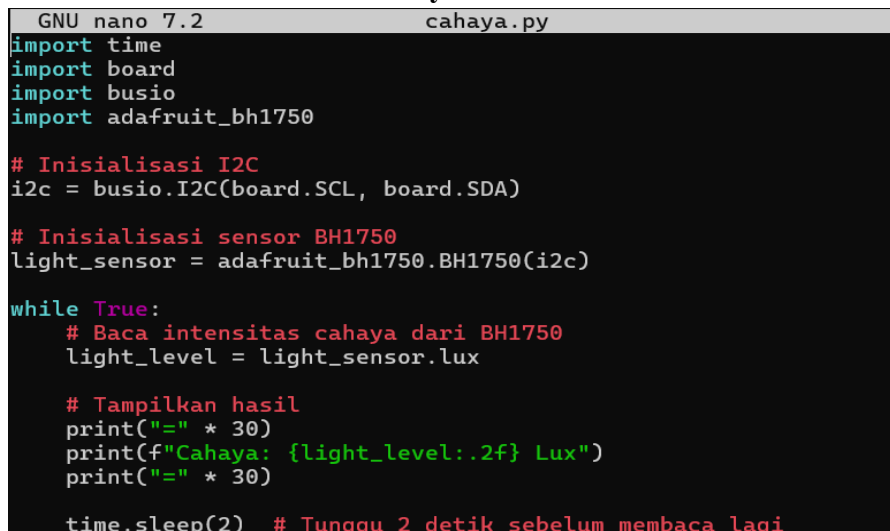
Program juga dilengkapi dengan exception handling untuk menangani interruption dari keyboard (Ctrl+C) yang akan menampilkan pesan "Dihentikan oleh pengguna", serta menangani error umum lainnya. Blok finally memastikan bahwa koneksi ke sensor ditutup dengan benar menggunakan `dhtDevice.exit()` ketika program berakhir, baik karena dihentikan secara normal maupun karena terjadi error. Struktur ini menjamin program berjalan stabil dan resources dibersihkan dengan proper saat program berakhir.

Berikut ini adalah hasil menjalankan program `suhu.py`



```
sitimarfungah@umygreenho x + v
(myenv) sitimarfungah@umygreenhouse:~ $ python3 suhu.py
Mulai membaca suhu dan kelembaban dari DHT22...
Tekan Ctrl+C untuk berhenti.
Suhu: 21.5°C Kelembaban: 49.4%
Suhu: 21.5°C Kelembaban: 49.3%
Suhu: 21.5°C Kelembaban: 49.3%
Suhu: 21.4°C Kelembaban: 49.3%
Suhu: 21.5°C Kelembaban: 49.3%
Suhu: 21.4°C Kelembaban: 49.3%
```

## B. BH1750 untuk Intensitas Cahaya



```
GNU nano 7.2 cahaya.py
import time
import board
import busio
import adafruit_bh1750

# Inisialisasi I2C
i2c = busio.I2C(board.SCL, board.SDA)

# Inisialisasi sensor BH1750
light_sensor = adafruit_bh1750.BH1750(i2c)

while True:
    # Baca intensitas cahaya dari BH1750
    light_level = light_sensor.lux

    # Tampilkan hasil
    print("=" * 30)
    print(f"Cahaya: {light_level:.2f} Lux")
    print("=" * 30)

    time.sleep(2) # Tunggu 2 detik sebelum membaca lagi
```

Code ini adalah program Python untuk membaca sensor cahaya BH1750 menggunakan protokol komunikasi I2C. Program dimulai dengan mengimpor library yang diperlukan yaitu time untuk delay, board untuk akses pin, busio untuk komunikasi I2C, dan adafruit\_bh1750 sebagai driver khusus sensor BH1750. Komunikasi I2C diinisialisasi dengan menggunakan pin SCL dan SDA dari board, kemudian sensor cahaya dibuat sebagai objek yang berkomunikasi melalui bus I2C tersebut.

Program kemudian masuk ke dalam loop tak terbatas yang secara kontinyu membaca intensitas cahaya dari sensor. Setiap iterasi, program mengambil nilai pencahayaan dalam satuan Lux melalui properti light\_sensor.lux, lalu menampilkannya dalam format yang rapi dengan pembatas garis menggunakan karakter "=" sebanyak 30 kali. Data cahaya ditampilkan dengan dua angka desimal untuk memberikan presisi yang cukup dalam pembacaan intensitas cahaya.

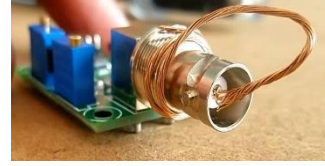
Setelah menampilkan hasil pembacaan, program memberikan jeda selama 2 detik sebelum melakukan pembacaan berikutnya menggunakan time.sleep(2). Struktur program ini sangat sederhana namun efektif untuk monitoring cahaya secara real-time, cocok untuk aplikasi seperti sistem pencahayaan otomatis, monitoring kondisi lingkungan, atau eksperimen pembelajaran tentang sensor cahaya dan komunikasi I2C.

### **C. pH Probe dengan Modul BNC untuk mengukur pH Tanah**

Sensor pH tanah perlu dikalibrasi untuk memastikan akurasi pengukuran. Berikut langkah-langkah kalibrasinya:

1. Siapkan larutan buffer pH 4.01, 6.86, dan 9.18

2. Lepaskan probe dari BNC nya
3. Gulung kawat di bagian konektor BNC dan pastikan kawat menyentuh bagian tengah konektor
4. jalankan kode berikut untuk memeriksa nilai tegangan :



```
GNU nano 7.2                                tegangan.py
import time
import board
import busio
import adafruit_ads1x15.ads1115 as ADS
from adafruit_ads1x15.analog_in import AnalogIn

# Setup I2C
i2c = busio.I2C(board.SCL, board.SDA)
ads = ADS.ADS1115(i2c)

# Pakai channel 0 (A0)
chan = AnalogIn(ads, ADS.P0)

print("Mulai baca tegangan...")
while True:
    print(f"Voltage: {chan.voltage:.3f} V")
    time.sleep(5)
```

Code ini adalah program Python untuk membaca tegangan analog menggunakan chip ADC (Analog-to-Digital Converter) ADS1115 melalui komunikasi I2C. Program mengimpor library yang diperlukan termasuk time, board, busio untuk I2C, serta library khusus adafruit\_ads1x15 yang menyediakan driver untuk chip ADS1115 dan kelas AnalogIn untuk membaca input analog. Chip ADS1115 digunakan karena banyak mikrocontroller seperti Raspberry Pi tidak memiliki pin ADC built-in, sehingga memerlukan chip eksternal untuk membaca sinyal analog.

Setelah menginisialisasi komunikasi I2C menggunakan pin SCL dan SDA, program membuat objek ADS1115 dan mengkonfigurasi channel analog input pada pin P0 (channel 0). Channel ini kemudian dijadikan objek AnalogIn yang memungkinkan pembacaan tegangan secara

langsung. Program menampilkan pesan pembuka kemudian masuk ke loop tak terbatas yang terus membaca dan menampilkan nilai tegangan.

Di dalam loop, program membaca tegangan melalui properti `chan.voltage` dan menampilkannya dengan format tiga angka desimal untuk presisi yang baik. Setiap pembacaan dipisahkan dengan delay 5 detik menggunakan `time.sleep(5)`. Program ini sangat berguna untuk monitoring tegangan dari berbagai sensor analog seperti potensiometer, sensor suhu analog, atau sensor tekanan yang menghasilkan output berupa tegangan bervariasi.

5. Pastikan angka tegangan sekitar 2.5V. jika terlalu besar, putar potensiometer offset dengan obeng kecil.
6. Pasang kembali pH probe ke konektor BNC
7. Lakukan kalibrasi dengan larutan buffer:
  - a. Masukkan pH probe ke larutan dengan pH 4.01. catat nilai tegangan
  - b. Bilas dan lap pH probe,
  - c. Masukkan pH probe ke larutan dengan pH 6.86. catat nilai tegangan
  - d. Bilas dan lap pH probe,
  - e. Masukkan pH probe ke larutan dengan pH 9.18. catat nilai tegangan
8. Buat file dengan nama `tanah.py` lalu masukkan kode berikut



```

GNU nano 7.2                                tanah.py *
import time
import board
import busio
import adafruit_ads1x15.ads1115 as ADS
from adafruit_ads1x15.analog_in import AnalogIn

# --- Setup I2C dan ADC ---
i2c = busio.I2C(board.SCL, board.SDA)
ads = ADS.ADS1115(i2c)
chan = AnalogIn(ads, ADS.P0) # pakai channel A0

# --- Nilai referensi hasil kalibrasi ---
tegangan_pH4 = 3.2 # contoh, ganti dengan hasil kamu pas kalibrasi pH 4
tegangan_pH7 = 2.52 # contoh, ganti dengan hasil kamu pas kalibrasi pH 7

# Hitung Ph_step
Ph_step = (tegangan_pH4 - tegangan_pH7) / (7.0 - 4.0)

# --- Loop utama ---
while True:
    voltage = chan.voltage # baca tegangan
    print("Tegangan: {:.3f} V".format(voltage))

    # Hitung pH
    pH = 7.00 + ((tegangan_pH7 - voltage) / Ph_step)

    print("pH cairan: {:.2f}".format(pH))
    print("-----")

    time.sleep(10) # delay 10 detik

```

Code ini adalah program Python untuk membaca sensor pH tanah menggunakan ADC ADS1115 dengan sistem kalibrasi untuk mengkonversi tegangan menjadi nilai pH yang akurat. Program dimulai dengan setup yang sama seperti pembacaan tegangan biasa, namun ditambahkan dengan sistem kalibrasi dua titik menggunakan larutan buffer pH 4 dan pH 7. Nilai referensi kalibrasi disimpan dalam variabel tegangan\_pH4 (3.2V) dan tegangan\_pH7 (2.52V) yang diperoleh dari hasil pengukuran sebelumnya saat sensor dicelupkan ke larutan buffer standar.

Program kemudian menghitung nilai Ph\_step yang merupakan gradien atau kemiringan kurva kalibrasi dengan rumus  $(\text{tegangan\_pH4} - \text{tegangan\_pH7}) / (7.0 - 4.0)$ . Nilai ini menunjukkan seberapa besar perubahan tegangan untuk setiap unit perubahan pH, yang penting untuk akurasi konversi. Di dalam loop utama, program membaca tegangan dari

sensor dan menampilkannya, kemudian menggunakan rumus kalibrasi linear untuk mengkonversi tegangan menjadi nilai pH.

Rumus konversi pH menggunakan persamaan  $7.00 + ((\text{tegangan\_pH7} - \text{voltage}) / \text{Ph\_step})$  yang merupakan interpolasi linear berdasarkan titik kalibrasi pH 7. Program menampilkan hasil pH dengan dua angka desimal dan memberikan delay 10 detik antar pembacaan. Metode kalibrasi dua titik ini memberikan akurasi yang baik untuk pengukuran pH dalam rentang normal, meskipun untuk akurasi maksimal biasanya diperlukan kalibrasi tiga titik dengan menambahkan buffer pH 10.

## 4.2 Pengujian Pengiriman Data

```
GNU nano 7.2 datasensor.py
import time
import board
import busio
import requests
import datetime
from adafruit_ads1x15.ads1115 import ADS1115, P0
from adafruit_ads1x15.analog_in import AnalogIn
import adafruit_dht
import adafruit_bh1750

# Setup I2C
i2c = busio.I2C(board.SCL, board.SDA)

# Inisialisasi sensor ADS1115
ads = ADS1115(i2c)
ph_channel = AnalogIn(ads, P0) # Sensor pH pakai channel A0 (P0)
soil_channel = AnalogIn(ads, P0) # Ganti ini ke P1 kalau soil sensor pakai A1

# Sensor DHT22
dht_device = adafruit_dht.DHT22(board.D4)

# Sensor cahaya BH1750
bh1750 = adafruit_bh1750.BH1750(i2c)

# Tegangan referensi hasil kalibrasi pH
tegangan_pH4 = 3.20
tegangan_pH7 = 2.52
Ph_step = (tegangan_pH4 - tegangan_pH7) / (7.0 - 4.0)

def hitung_pH(tegangan):
    return round(7.00 + ((tegangan_pH7 - tegangan) / Ph_step), 2)

url = "https://ilmuku.id/datalogger/store"
headers = {"Content-Type": "application/x-www-form-urlencoded"}
```

```

def main():
    counter = 1
    while True:
        try:
            now = datetime.datetime.now()
            current_date = now.strftime("%Y-%m-%d")
            hour = now.hour
            minute = 30 if now.minute >= 30 else 0
            current_time = f"{hour:02d}:{minute:02d}:00"

            try:
                suhu = dht_device.temperature
                kelembapan_udara = dht_device.humidity
            except:
                suhu = 25.0
                kelembapan_udara = 60.0

            try:
                cahaya = bh1750.lux
            except:
                cahaya = 150.0

            try:
                tegangan_ph = ph_channel.voltage
                ph_tanah = hitung_pH(tegangan_ph)
            except:
                tegangan_ph = 2.52
                ph_tanah = 7.00

            data = {
                "pid": "Stasiun01",
                "did": "Device01",
                "gdate": current_date,
                "gtime": current_time,
                "v1": str(round(suhu, 1)),
                "v2": str(round(kelembapan_udara, 1)),
                "v3": "0",
                "v4": str(round(cahaya, 1)),
                "v5": str(round(ph_tanah, 2)),
                "type": '{"pr':'1','n':'','p':'','k':''}'
            }

            response = requests.post(url, headers=headers, data=data)

            print(f"[{counter}] {current_date} {current_time}")
            print(f"  Suhu          : {suhu:.1f} °C")
            print(f"  Kelembapan Udara : {kelembapan_udara:.1f} %")
            print(f"  Cahaya         : {cahaya:.1f} Lux")
            print(f"  Tegangan pH      : {tegangan_ph:.3f} V")
            print(f"  pH Tanah        : {ph_tanah:.2f}")
            print(f"  Server response  : {response.status_code} {response.text}")
            print("-" * 60)

            counter += 1
            time.sleep(1800)

        except Exception as e:
            print(f"Error umum: {e}")
            time.sleep(10)

if __name__ == "__main__":
    print("Mulai membaca sensor dan kirim data ke server setiap 30 menit...")
    main()

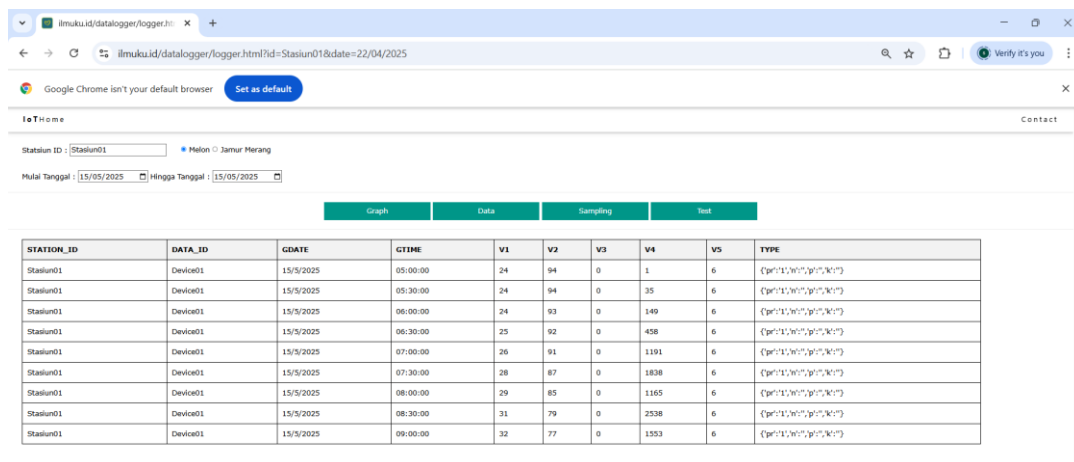
```

Code ini adalah program Python yang mengintegrasikan multiple sensor untuk membuat sistem data logger lengkap yang mengirim data ke server online. Program menggabungkan tiga sensor sekaligus yaitu DHT22 untuk suhu dan kelembaban, BH1750 untuk intensitas cahaya, dan sensor

pH tanah menggunakan ADC ADS1115. Semua sensor dikonfigurasi dengan setup I2C yang tepat, termasuk kalibrasi pH menggunakan metode dua titik seperti program sebelumnya.

Program memiliki fungsi `hitung_pH` yang melakukan konversi tegangan ke nilai pH dengan akurasi dua desimal, dan fungsi `main` yang menjalankan loop pengambilan data secara kontinyu. Di dalam loop, program menggunakan `datetime` untuk mencatat waktu pengambilan data dengan format yang rapi, kemudian secara berurutan membaca semua sensor dengan exception handling untuk setiap sensor. Jika sensor gagal dibaca, program akan menggunakan nilai default yang masuk akal sehingga sistem tetap berjalan meskipun ada sensor yang bermasalah.

Data dari semua sensor kemudian dikemas dalam dictionary JSON yang terstruktur dengan field `pid` untuk identifikasi stasiun, `did` untuk device ID, tanggal dan waktu pengambilan data, serta nilai-nilai sensor (`v1` untuk suhu, `v2` untuk kelembaban, `v3` untuk intensitas cahaya, dan `v5` untuk pH tanah). Program menggunakan `library requests` untuk mengirim data ke server melalui HTTP POST ke URL yang telah ditentukan, menjadikan ini sebagai sistem IoT monitoring lingkungan yang lengkap dan dapat diakses secara remote untuk keperluan monitoring pertanian atau lingkungan.



STATION_ID	DATA_ID	GDATE	GTIME	V1	V2	V3	V4	V5	TYPE
Stasiun01	Device01	15/5/2025	05:00:00	24	94	0	1	6	{"v1":24,"v2":94,"v3":0,"v4":1,"v5":6,"date":"15/5/2025 05:00:00","station":"Stasiun01","device":"Device01","type":"data"}
Stasiun01	Device01	15/5/2025	05:30:00	24	94	0	35	6	{"v1":24,"v2":94,"v3":0,"v4":35,"v5":6,"date":"15/5/2025 05:30:00","station":"Stasiun01","device":"Device01","type":"data"}
Stasiun01	Device01	15/5/2025	06:00:00	24	93	0	149	6	{"v1":24,"v2":93,"v3":0,"v4":149,"v5":6,"date":"15/5/2025 06:00:00","station":"Stasiun01","device":"Device01","type":"data"}
Stasiun01	Device01	15/5/2025	06:30:00	25	92	0	458	6	{"v1":25,"v2":92,"v3":0,"v4":458,"v5":6,"date":"15/5/2025 06:30:00","station":"Stasiun01","device":"Device01","type":"data"}
Stasiun01	Device01	15/5/2025	07:00:00	26	91	0	1191	6	{"v1":26,"v2":91,"v3":0,"v4":1191,"v5":6,"date":"15/5/2025 07:00:00","station":"Stasiun01","device":"Device01","type":"data"}
Stasiun01	Device01	15/5/2025	07:30:00	28	87	0	1838	6	{"v1":28,"v2":87,"v3":0,"v4":1838,"v5":6,"date":"15/5/2025 07:30:00","station":"Stasiun01","device":"Device01","type":"data"}
Stasiun01	Device01	15/5/2025	08:00:00	29	85	0	1165	6	{"v1":29,"v2":85,"v3":0,"v4":1165,"v5":6,"date":"15/5/2025 08:00:00","station":"Stasiun01","device":"Device01","type":"data"}
Stasiun01	Device01	15/5/2025	08:30:00	31	79	0	2538	6	{"v1":31,"v2":79,"v3":0,"v4":2538,"v5":6,"date":"15/5/2025 08:30:00","station":"Stasiun01","device":"Device01","type":"data"}
Stasiun01	Device01	15/5/2025	09:00:00	32	77	0	1553	6	{"v1":32,"v2":77,"v3":0,"v4":1553,"v5":6,"date":"15/5/2025 09:00:00","station":"Stasiun01","device":"Device01","type":"data"}

Interface web ini adalah dashboard untuk sistem monitoring sensor lingkungan yang menampilkan data dari stasiun IoT yang telah dikirim melalui program Python sebelumnya. Dashboard menampilkan data dalam bentuk tabel yang terorganisir dengan baik, dimana setiap baris mewakili satu record pengambilan data dari sensor pada waktu tertentu. Kolom-kolom yang ditampilkan meliputi STATION\_ID untuk identifikasi stasiun (Stasiun01), DATA\_ID sebagai unique identifier setiap record, serta kolom GDATE dan GTIME yang menunjukkan tanggal dan waktu pengambilan data.

Data sensor ditampilkan dalam kolom V1 hingga V5 yang masing-masing merepresentasikan nilai dari sensor berbeda sesuai dengan struktur JSON yang dikirim program Python. V1 menampilkan suhu dalam derajat Celsius, V2 untuk kelembaban udara dalam persen, V3 untuk intensitas cahaya dalam Lux, V4 tampaknya kosong atau tidak digunakan, dan V5 untuk nilai pH tanah. Interface ini juga dilengkapi dengan fitur filter tanggal di bagian atas yang memungkinkan pengguna memilih rentang waktu tertentu untuk melihat data historis.

Dashboard ini sangat berguna untuk monitoring jangka panjang kondisi lingkungan, memungkinkan analisis tren data dari waktu ke waktu, dan memberikan kemudahan akses data dari mana saja melalui web browser. Pengguna dapat dengan mudah melihat pola perubahan suhu, kelembaban, intensitas cahaya, dan pH tanah yang penting untuk aplikasi seperti smart farming, monitoring cuaca, atau penelitian lingkungan. Data yang tersimpan secara online ini juga memungkinkan multiple user untuk mengakses informasi yang sama secara bersamaan.

Stasiun ID : Stasiun01

● Melon ○ Jamur Merang

Mulai Tanggal : 07/05/2025



Hingga Tanggal : 10/05/2025



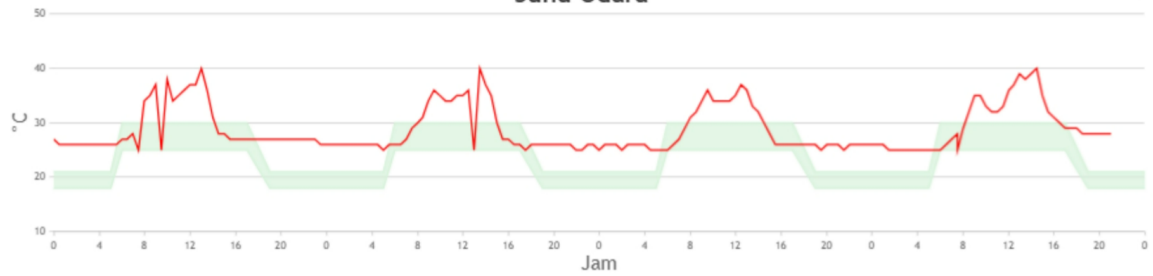
Graph

Data

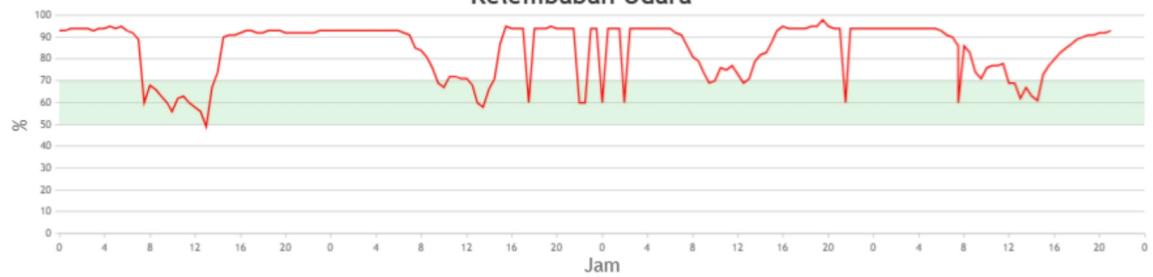
Sampling

Test

## Suhu Udara



## Kelembaban Udara



## Intensitas Cahaya



## pH Tanah



Contact

Information Technology Department  
Universitas Muhammadiyah Yogyakarta  
Jl. Brawijaya Tamantirto Bantul DI Yogyakarta - 55183  
email: purbohadi@yahoo.com

Stasiun ID : Stasiun01

○ Melon ● Jamur Merang

Mulai Tanggal : 07/05/2025



Hingga Tanggal : 10/05/2025



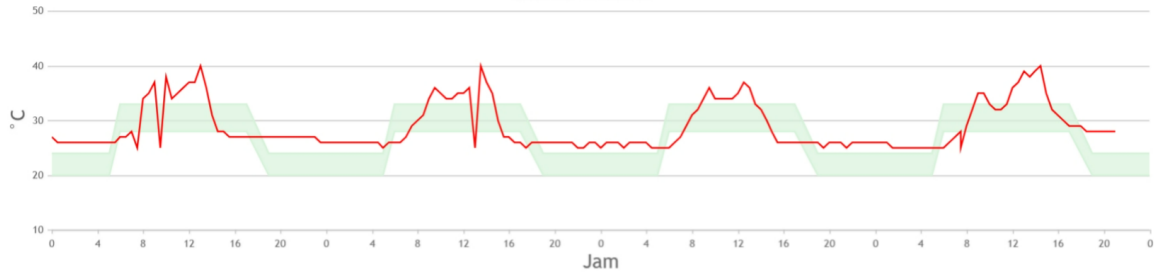
Graph

Data

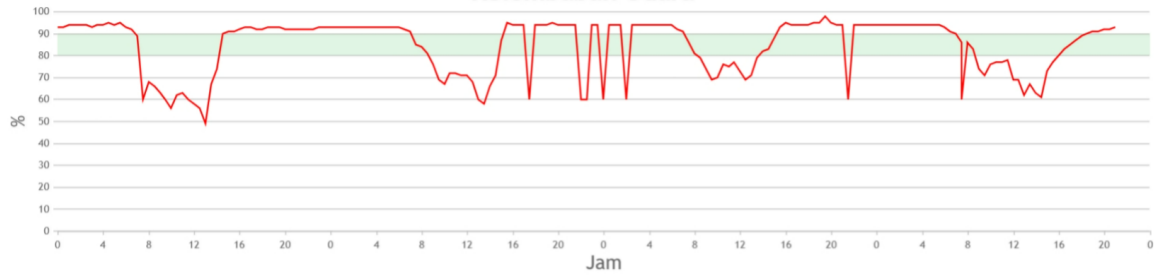
Sampling

Test

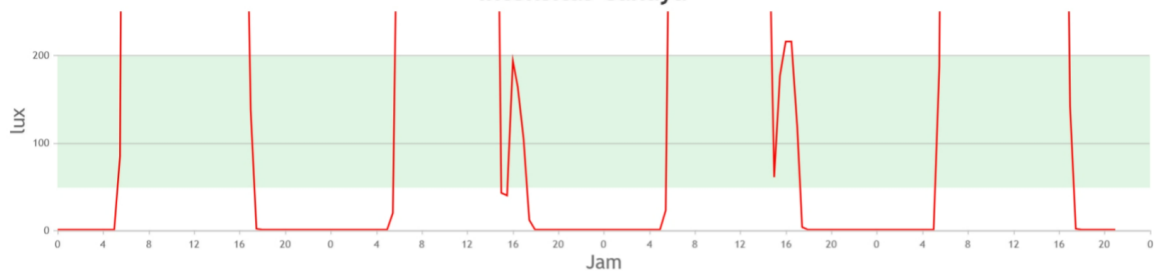
## Suhu Udara



## Kelembaban Udara



## Intensitas Cahaya



## pH Tanah



Contact

Information Technology Department  
Universitas Muhammadiyah Yogyakarta  
Jl. Brawijaya Tamantirto Bantul DI Yogyakarta - 55183  
email: purbohadi@yahoo.com

### **Sistem Pengiriman Data dan Monitoring**

Sistem monitoring lingkungan ini dirancang untuk mengirimkan data sensor secara otomatis setiap 30 menit ke server online. Setiap kali data berhasil dikirim, sistem akan menampilkan konfirmasi di terminal untuk memastikan proses pengiriman berjalan dengan baik. Data yang terkumpul kemudian dapat diakses melalui dashboard web yang menyediakan visualisasi dalam bentuk grafik dan tabel.

### **Visualisasi Data dan Zona Optimal**

Dashboard monitoring menampilkan dua elemen penting dalam visualisasinya. Pertama adalah garis merah yang menunjukkan data aktual dari pembacaan sensor secara real-time. Kedua adalah area hijau yang merepresentasikan rentang nilai optimal untuk parameter lingkungan tertentu, seperti suhu ideal untuk pertumbuhan tanaman atau kondisi lingkungan yang diinginkan. Fitur zona hijau ini membantu pengguna untuk dengan cepat mengidentifikasi apakah kondisi lingkungan berada dalam rentang yang sesuai.

### **Contoh Rentang Nilai Optimal**

Sebagai ilustrasi, berikut adalah contoh rentang nilai optimal untuk dua jenis tanaman yang berbeda. Untuk tanaman melon, suhu udara optimal berkisar 25-30°C dengan kelembaban 50-70%, intensitas cahaya 25000-40000 Lux, dan pH tanah 6.0-7.0.

Sementara untuk jamur merang membutuhkan suhu yang sedikit lebih tinggi yaitu 28-33°C, kelembaban yang lebih tinggi 80-90%, intensitas cahaya yang rendah 50-200 Lux, dan pH tanah 6.5-7.5. Perbedaan kebutuhan ini menunjukkan pentingnya monitoring parameter yang tepat sesuai dengan jenis tanaman yang dibudidayakan.

### **Penanganan Error dan Stabilitas Sistem**

Sistem dilengkapi dengan mekanisme penanganan error yang canggih menggunakan blok try-except dalam pemrograman. Ketika salah satu sensor mengalami kegagalan pembacaan, sistem tidak akan berhenti beroperasi melainkan menggunakan nilai default yang masuk akal seperti suhu 25°C, kelembaban 60%, intensitas cahaya 150 Lux, dan pH 7.00. Strategi ini memastikan kontinuitas pencatatan data meskipun terkadang



menghasilkan data anomali yang terlihat sebagai lompatan nilai yang tidak wajar dalam grafik. Mekanisme ini sangat penting untuk menjaga keandalan sistem monitoring jangka panjang, terutama dalam aplikasi yang memerlukan data kontinyu seperti smart farming atau monitoring lingkungan industri.

## BAB V. MONITORING JARAK JAUH

Monitoring jarak jauh memungkinkan pengguna untuk mengakses Raspberry Pi dan data sensor dari mana saja tanpa harus hadir langsung di lokasi greenhouse. Hal ini sangat penting untuk fleksibilitas pembelajaran dan perawatan sistem. Tailscale adalah layanan VPN yang memungkinkan koneksi antar perangkat melalui jaringan pribadi yang aman. Setiap perangkat diberi alamat IP virtual sehingga bisa saling terhubung dalam satu jaringan lokal, tanpa konfigurasi jaringan yang rumit.

Langkah-langkah instalasi dan konfigurasi adalah sebagai berikut :

1. Buat file konfigurasi Jaringan

```
sudo nano /etc/wpa_supplicant/wpa_supplicant.conf
```

2. Masukkan konfigurasi berikut (Sesuaikan id, ssid, dan psk dengan modem masing-masing)

```
[connection]
id=4G-UFI-044c
uuid=8f62e2a5-3ba5-4cf4-8ee7-62133ed204e4
type=wifi
interface-name=wlan0

[wifi]
mode=infrastructure
ssid=4G-UFI-044c

[wifi-security]
auth-alg=open
key-mgmt=wpa-psk
psk=1234567890

[ipv4]
method=auto

[ipv6]
addr-gen-mode=default
method=auto

[proxy]
```

## **Konfigurasi Jaringan Sistem IoT**

File konfigurasi ini menunjukkan pengaturan koneksi jaringan untuk sistem IoT yang menggunakan koneksi WiFi sebagai media komunikasi data. Konfigurasi dimulai dengan bagian [connection] yang berisi identitas unik perangkat melalui parameter id dan uuid, serta menentukan tipe koneksi yang digunakan yaitu WiFi dengan interface wlan0 yang merupakan interface jaringan nirkabel standar pada sistem Linux.

Bagian [wifi] mengatur mode operasi perangkat sebagai infrastructure, yang berarti perangkat akan terhubung ke access point WiFi sebagai client, bukan sebagai hotspot. Parameter ssid menunjukkan nama jaringan WiFi yang akan digunakan untuk koneksi, dalam hal ini adalah "4G-UFI-044c" yang tampaknya merupakan mobile hotspot atau router portabel dengan koneksi 4G.

## **Konfigurasi Keamanan dan Protokol Jaringan**

Bagian [wifi-security] menunjukkan pengaturan keamanan jaringan yang menggunakan algoritma autentikasi terbuka (open) tanpa enkripsi WPA atau WEP. Parameter key-mgmt diset sebagai wpa-psk yang mengindikasikan penggunaan WPA Pre-Shared Key, dengan nilai psk berupa kunci numerik untuk autentikasi. Konfigurasi IPv4 dan IPv6 keduanya diatur dalam mode otomatis, yang berarti sistem akan secara otomatis mendapatkan alamat IP dari DHCP server yang tersedia di jaringan.

Bagian [proxy] yang kosong menunjukkan bahwa sistem tidak menggunakan proxy server untuk koneksi internet, sehingga komunikasi data dilakukan secara langsung ke server tujuan. Konfigurasi ini cocok untuk sistem IoT yang memerlukan koneksi internet stabil untuk mengirim data sensor ke cloud platform, dengan menggunakan mobile hotspot sebagai gateway internet yang dapat diandalkan di lokasi yang tidak memiliki infrastruktur WiFi permanen.

3. Simpan dan restart layanan networking

```
sudo wpa_cli -i wlan0 reconfigure  
sudo systemctl restart networking
```

## **Instalasi dan Konfigurasi Tailscale VPN melalui browser**

- a. Instalasi Tailscale di Raspberry Pi (Server)

1. Buka Terminal di Raspberry Pi dan install Tailscale

```
sudo wpa_cli -i wlan0 reconfigure  
sudo systemctl restart networking
```

2. Setelah instalasi selesai, jalankan perintah berikut untuk menghasilkan URL autentikasi

```
sudo tailscale up
```

3. Perintah ini akan menampilkan URL seperti berikut

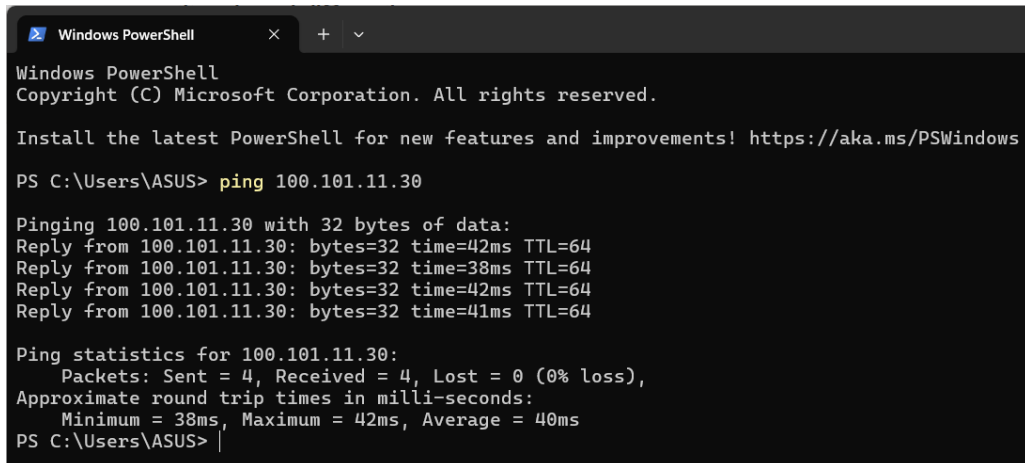
```
To authenticate, visit:  
  
https://login.tailscale.com/a/abcdef123456
```

- b. Konfigurasi Tailscale di Laptop/Komputer Client Melalui Browser

1. Buka browser di laptop/komputer client
2. Kunjungi <https://login.tailscale.com>
3. Login menggunakan akun yang sama yang digunakan di Raspberry Pi
4. Setelah login, klik tombol "Download Tailscale" di sudut kanan atas
5. Pilih versi untuk sistem operasi laptop/komputer (Windows, macOS, Linux)
6. Download dan instal aplikasi Tailscale
7. Setelah instalasi selesai, aplikasi Tailscale akan meminta untuk login
8. Pilih "Log in from browser" atau "Login dari browser"
9. Browser akan terbuka secara otomatis dengan halaman autentikasi Tailscale
10. Karena Anda sudah login sebelumnya, perangkat akan otomatis dikenali dan terhubung

11. Berikan nama yang mudah diingat untuk perangkat client, misalnya "Laptop-Monitoring"
  12. Klik "Approve" atau "Setujui" untuk menyelesaikan proses autentikasi
- c. Verifikasi Koneksi Tailscale Melalui Web Admin Panel
1. Di browser (baik di Raspberry Pi atau laptop), buka <https://login.tailscale.com/admin/machines>
  2. Login jika diminta
  3. Halaman ini akan menampilkan semua perangkat yang terhubung ke jaringan Tailscale Anda
  4. Verifikasi bahwa kedua perangkat (Raspberry Pi dan laptop) muncul dalam daftar dan statusnya "Online"
  5. Catat alamat IP Tailscale dari Raspberry Pi (biasanya dalam format 100.x.y.z)
  6. Alamat IP ini akan digunakan untuk mengakses Raspberry Pi dari laptop client
- d. Tes Ping

```
ping [raspberry_pi_tailscale_ip]
# Contoh: ping 100.x.y.z
```



```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\ASUS> ping 100.101.11.30

Pinging 100.101.11.30 with 32 bytes of data:
Reply from 100.101.11.30: bytes=32 time=42ms TTL=64
Reply from 100.101.11.30: bytes=32 time=38ms TTL=64
Reply from 100.101.11.30: bytes=32 time=42ms TTL=64
Reply from 100.101.11.30: bytes=32 time=41ms TTL=64

Ping statistics for 100.101.11.30:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 38ms, Maximum = 42ms, Average = 40ms
PS C:\Users\ASUS> |
```

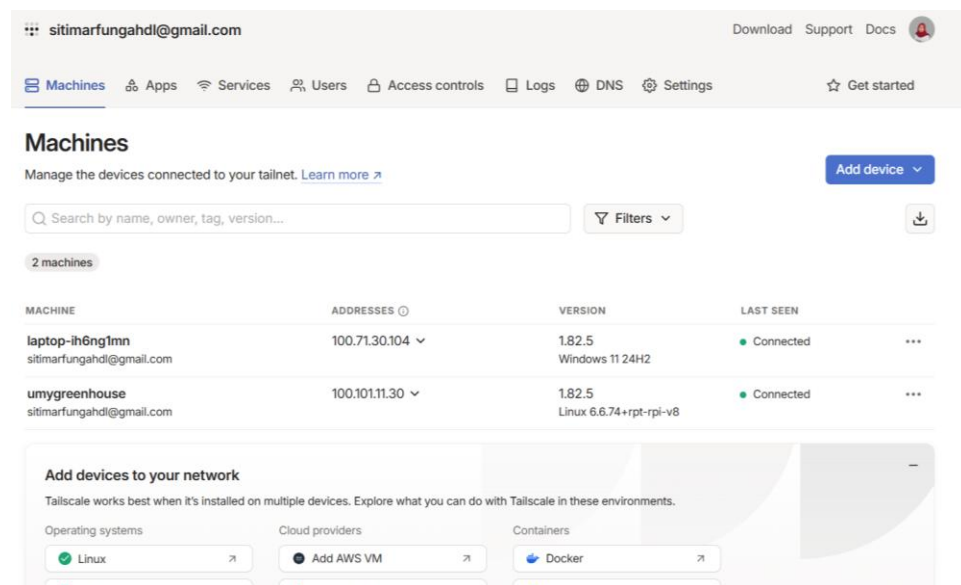
- e. Mengakses Raspberry Pi dari Jarak Jauh melalui SSH

```
ssh [username]@[raspberry_pi_tailscale_ip]
# Contoh: ssh pi@100.x.y.z
# atau menggunakan hostname
ssh pi@raspberrypi
```

```
PS C:\Users\ASUS> ssh sitimarfungah@100.101.11.30
sitimarfungah@100.101.11.30's password:
Linux umygreenhouse 6.6.74+rpt-rpi-v8 #1 SMP PREEMPT Debian 1:6.6.74-1+rpt1 (2025-01-27) aarch64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Fri May 30 14:04:28 2025 from 100.71.30.104
sitimarfungah@umygreenhouse:~$
```



## Dashboards Manajemen Perangkat IoT dengan Tailscale

Interface ini menunjukkan dashboard Tailscale yang berfungsi sebagai sistem manajemen jaringan VPN untuk perangkat IoT. Dashboard menampilkan dua perangkat yang terhubung dalam jaringan, yaitu "laptop-ih6ng1mn" dengan sistem operasi Windows 11 dan "umygreenhouse" yang menjalankan Linux. Kedua perangkat memiliki status "Connected" yang menunjukkan koneksi aktif ke jaringan Tailscale dengan alamat IP internal masing-masing 100.71.30.104 dan 100.101.11.30.

Tailscale menggunakan teknologi mesh VPN yang memungkinkan perangkat IoT untuk berkomunikasi secara aman melalui internet tanpa perlu konfigurasi port forwarding atau firewall yang rumit. Setiap perangkat mendapatkan alamat IP unik dalam rentang 100.x.x.x yang merupakan karakteristik jaringan Tailscale, memungkinkan akses remote yang aman ke sistem monitoring dari mana saja selama terhubung internet.

### **Fitur Manajemen dan Skalabilitas Sistem**

Dashboard menyediakan fitur pencarian, filter, dan opsi untuk menambahkan perangkat baru ke jaringan melalui tombol "Add device". Bagian bawah interface menunjukkan opsi untuk mengintegrasikan berbagai platform seperti sistem operasi Linux dan Windows, cloud provider seperti AWS, serta containerization dengan Docker. Hal ini menunjukkan fleksibilitas Tailscale dalam mendukung deployment sistem IoT yang kompleks dan dapat diskalakan sesuai kebutuhan.

Penggunaan Tailscale dalam sistem IoT monitoring memberikan keuntungan berupa kemudahan akses remote ke perangkat sensor, keamanan komunikasi data melalui enkripsi end-to-end, dan kemampuan untuk mengelola multiple site monitoring dari satu dashboard terpusat. Sistem ini sangat cocok untuk aplikasi smart farming atau industrial IoT yang memerlukan monitoring dari lokasi yang tersebar geografis dengan tetap menjaga keamanan data.