

**LAPORAN UTS**  
**WIRELESS SENSOR NETWORK**



Disusun Oleh:

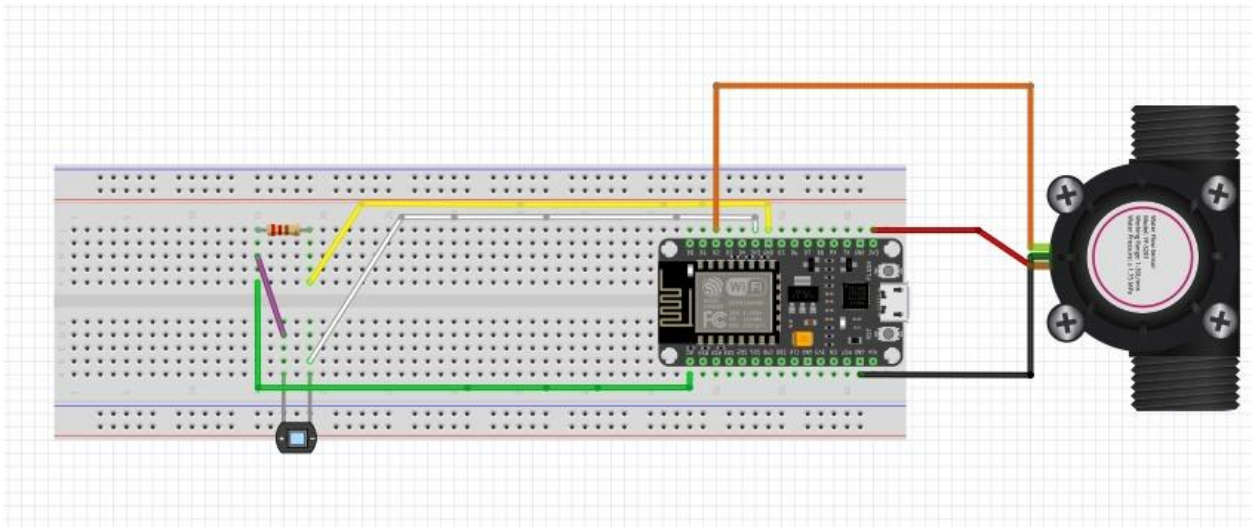
Alif Rendina Pamungan	(2207421034)
Ibrahim Alvaro	(2207421035)
M. Berryl Muchtada	(2207421045)
Fahrul Rosi	(2207421052)

Dosen Pengampu:

Dr. Prihatin Oktivasari, M.Si

**PROGRAM STUDI D4 TEKNIK MULTIMEDIA DAN JARINGAN**  
**JURUSAN TEKNIK INFORMATIKA DAN KOMPUTER**  
**POLITEKNIK NEGERI JAKARTA**  
**TAHUN 2023/2024**

## 1. DESAIN HARDWARE



Berikut adalah penjelasan dari skematik rangkaian pada gambar di atas:

### 1.) Komponen yang digunakan:

- Breadboard
- Microcontroller ESP8266 (NodeMCU)
- Sensor aliran air
- Resistor
- Kabel jumper
- Photodiode

### 2.) Rangkaian:

- Sensor Aliran Air:
  - Sensor aliran air memiliki tiga kabel: merah (VCC), kuning (Data), dan hitam (GND).
  - Kabel merah (VCC) sensor terhubung ke pin 3.3V pada ESP8266.
  - Kabel hitam (GND) sensor terhubung ke pin GND pada ESP8266.

- Kabel kuning (Data) sensor terhubung ke pin D2 (GPIO4) pada ESP8266 melalui resistor.
- Photodiode:
  - Photodiode digunakan untuk mengukur intensitas cahaya pada air. Umumnya, photodiode memiliki dua terminal: anoda dan katoda.
  - Anoda photodiode dihubungkan ke VCC (3.3V) melalui resistor.
  - Katoda photodiode dihubungkan ke pin GND.
  - Sinyal keluaran dari photodiode dihubungkan ke salah satu pin analog pada ESP8266 (pin A0) untuk membaca nilai intensitas cahaya.

### 3.) Resistor:

- Resistor digunakan dalam dua tempat:
  - Satu resistor ditempatkan antara pin data sensor aliran air dan pin GPIO pada ESP8266.
  - Satu resistor digunakan dengan photodiode untuk membentuk pembagi tegangan, sehingga sinyal yang diterima pada pin analog ESP8266 dapat terbaca dengan baik.

### 4.) Kabel Jumper:

- Kabel jumper digunakan untuk menghubungkan semua komponen sesuai dengan koneksi yang dijelaskan di atas.

### 5.) ESP8266 (NodeMCU):

- NodeMCU adalah platform open-source IoT yang terdiri dari perangkat keras dan firmware yang dapat di-flash pada ESP8266.

- NodeMCU memiliki beberapa pin GPIO yang dapat digunakan untuk berbagai fungsi input dan output, serta satu pin analog (A0).

Fungsi Rangkaian:

1. Membaca data aliran air menggunakan sensor aliran air dan mengirimkan data tersebut ke microcontroller ESP8266.
2. Mengukur intensitas cahaya pada air menggunakan photodiode dan mengirimkan data tersebut ke pin analog ESP8266.

## 2. SOURCE CODE

```
#define BLYNK_TEMPLATE_ID "TMPL6NASlouf1"
#define BLYNK_TEMPLATE_NAME "Kelompok 5"
#define BLYNK_AUTH_TOKEN "HCPfc3C4QblA-6ulljBIOBRC73he4u7B"
```

- **#define:** Mendefinisikan makro untuk ID template Blynk, nama template, dan token autentikasi yang digunakan untuk menghubungkan perangkat dengan aplikasi Blynk.

```
#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>
```

- **#include:** Mengikutkan library yang diperlukan. ESP8266WiFi.h untuk koneksi WiFi dan BlynkSimpleEsp8266.h untuk integrasi dengan Blynk.

```
char ssid[] = "Trimasketir";
char pass[] = "keluargaatung";
```

- **char ssid[] dan char pass[]:** Mendefinisikan SSID dan password WiFi yang akan digunakan untuk menghubungkan ESP8266 ke jaringan WiFi.

```
const int flowSensorPin = D2; // Water flow sensor pin

// Variables for pulse count and flow rate calculation
volatile int pulseCount = 0;
float flowRate = 0.0;
unsigned long oldTime = 0;
```

- **const int flowSensorPin:** Mendefinisikan pin untuk sensor aliran air.
- **volatile int pulseCount:** Variabel yang digunakan dalam interrupt service routine (ISR) untuk menghitung jumlah pulsa.

- **float flowRate:** Variabel untuk menyimpan laju aliran air.
- **unsigned long oldTime:** Variabel untuk menyimpan waktu sebelumnya dalam milidetik.

```
void IRAM_ATTR pulseCounter();
void calculateFlowRate();
void readLightIntensity();
```

- **void IRAM\_ATTR pulseCounter():** Deklarasi fungsi untuk menghitung pulsa dari sensor aliran air.
- **void calculateFlowRate():** Deklarasi fungsi untuk menghitung laju aliran air.
- **void readLightIntensity():** Deklarasi fungsi untuk membaca intensitas cahaya.

```
BlynkTimer timer;

void setup(void) {
  // Start serial communication
  Serial.begin(9600);
  Serial.println("Program dimulai...");

  // Initialize WiFi connection
  WiFi.begin(ssid, pass);
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  Serial.println("");
  Serial.println("WiFi connected");

  // Initialize Blynk
  Blynk.begin(BLYNK_AUTH_TOKEN, ssid, pass);

  // Initialize the flow sensor pin as input
  pinMode(flowSensorPin, INPUT);

  // Attach the interrupt service routine to the flow sensor pin
  attachInterrupt(digitalPinToInterrupt(flowSensorPin), pulseCounter, FALLING);

  // Set up a timer to calculate flow rate every second
  timer.setInterval(2000L, calculateFlowRate);

  // Set up a timer to read light intensity every 2 seconds
  timer.setInterval(2000L, readLightIntensity);
}
```

- **Serial.begin(9600):** Memulai komunikasi serial pada baud rate 9600.
- **WiFi.begin(ssid, pass):** Memulai koneksi ke jaringan WiFi menggunakan SSID dan password.
- **while (WiFi.status() != WL\_CONNECTED):** Menunggu hingga perangkat terhubung ke WiFi.

- **Blynk.begin(BLYNK\_AUTH\_TOKEN, ssid, pass):** Memulai koneksi ke Blynk menggunakan token autentikasi dan detail WiFi.
- **pinMode(flowSensorPin, INPUT):** Menetapkan pin sensor aliran sebagai input.
- **attachInterrupt(digitalPinToInterrupt(flowSensorPin), pulseCounter, FALLING):** Mengatur interrupt untuk menghitung pulsa dari sensor aliran air pada sinyal jatuh.
- **timer.setInterval(2000L, calculateFlowRate) dan timer.setInterval(2000L, readLightIntensity):** Mengatur interval waktu untuk menjalankan fungsi calculateFlowRate dan readLightIntensity setiap 2 detik.

```
void loop(void) {
  Blynk.run(); // Run Blynk
  timer.run(); // Run Blynk timer
}
```

- **Blynk.run():** Menjalankan proses Blynk.
- **timer.run():** Menjalankan timer yang telah diatur.

```
void IRAM_ATTR pulseCounter() {
  pulseCount++;
}
```

- **pulseCounter():** Fungsi yang akan dipanggil setiap kali ada sinyal jatuh pada pin sensor aliran air, menambah pulseCount.

```
void calculateFlowRate() {
  if ((millis() - oldTime) > 1000) {
    // Disable interrupts while calculating flow rate
    detachInterrupt(digitalPinToInterrupt(flowSensorPin));

    // Calculate flow rate in L/min
    flowRate = ((1000.0 / (millis() - oldTime)) * pulseCount) / 7.5;
    oldTime = millis();
    pulseCount = 0;

    // Print the flow rate to the serial monitor
    Serial.print("Flow rate: ");
    Serial.print(flowRate);
    Serial.println(" L/min");

    // Send flow rate to Blynk
    Blynk.virtualWrite(V2, flowRate);

    // Reattach the interrupt service routine
    attachInterrupt(digitalPinToInterrupt(flowSensorPin), pulseCounter, FALLING);
  }
}
```

- **calculateFlowRate():** Fungsi untuk menghitung laju aliran air dalam liter per menit (L/min).
- **detachInterrupt(digitalPinToInterrupt(flowSensorPin)):** Menonaktifkan interrupt sementara saat menghitung.
- **flowRate:** Menghitung laju aliran berdasarkan jumlah pulsa yang terdeteksi.
- **Serial.print():** Mencetak laju aliran ke serial monitor.
- **Blynk.virtualWrite(V2, flowRate):** Mengirimkan data laju aliran ke aplikasi Blynk pada virtual pin V2.
- **attachInterrupt(digitalPinToInterrupt(flowSensorPin), pulseCounter, FALLING):** Mengaktifkan kembali interrupt.

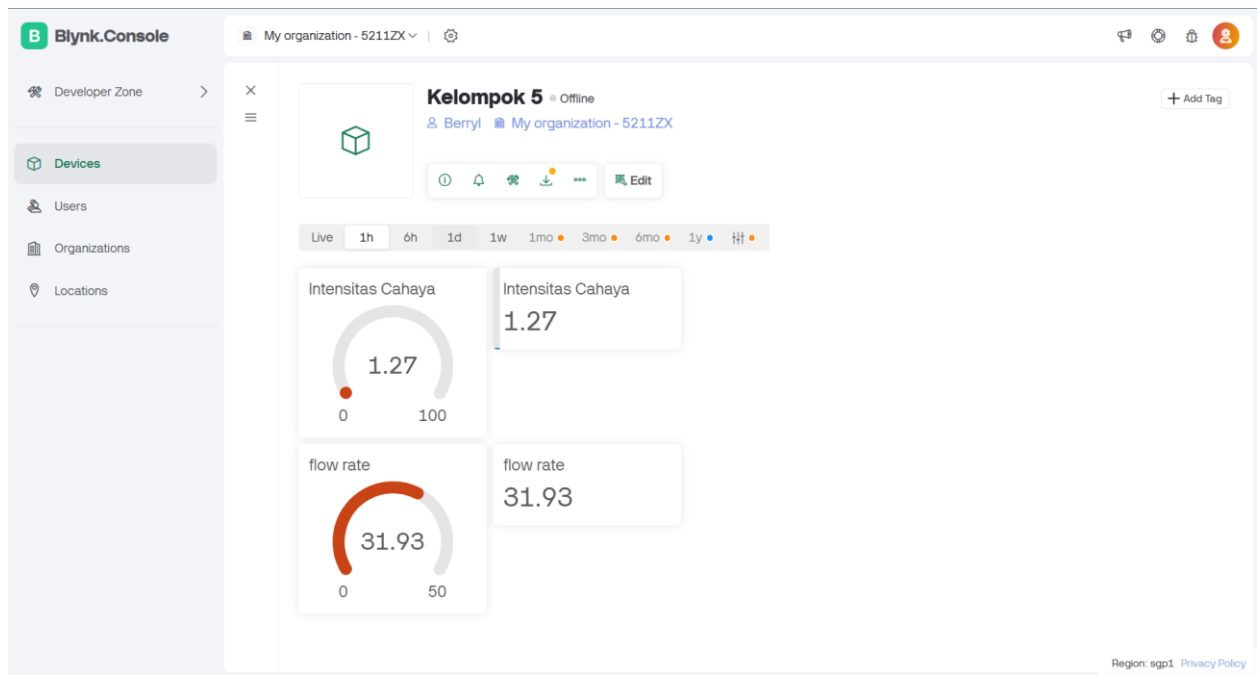
```
void readLightIntensity() {
  int pinRead0 = analogRead(A0); // Read analog pin A0
  float pVolt0 = pinRead0 / 1024.0 * 3.3; // Convert ADC value to voltage
  float persenCahaya = (pinRead0 / 1023.0) * 100; // Convert ADC value to percentage

  Serial.print("Nilai ADC: ");
  Serial.print(pinRead0); // Print ADC value
  Serial.print(" -> Tegangan: ");
  Serial.print(pVolt0); // Print voltage
  Serial.print(" V -> Intensitas Cahaya: ");
  Serial.print(persenCahaya); // Print light intensity percentage
  Serial.println(" ");

  // Send light intensity percentage to Blynk
  Blynk.virtualWrite(V0, persenCahaya);
}
```

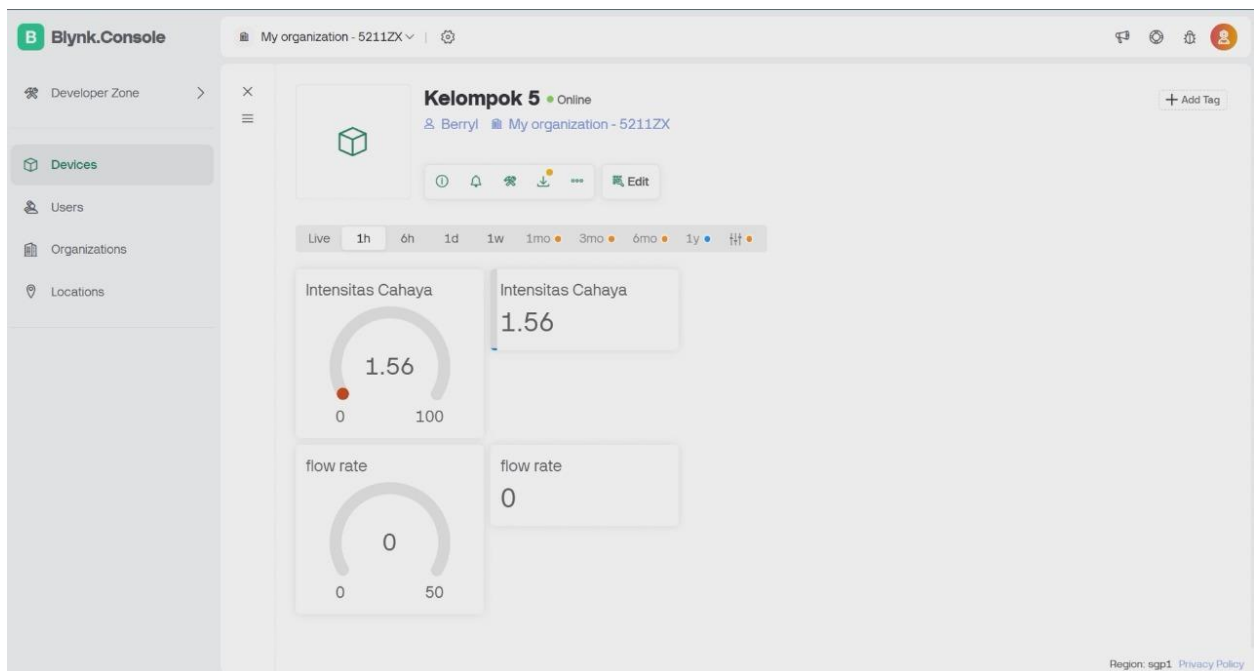
- **readLightIntensity():** Fungsi untuk membaca intensitas cahaya.
- **analogRead(A0):** Membaca nilai analog dari pin A0.
- **pVolt0:** Mengkonversi nilai ADC ke tegangan.
- **persenCahaya:** Mengkonversi nilai ADC ke persentase intensitas cahaya.
- **Serial.print():** Mencetak nilai ADC, tegangan, dan intensitas cahaya ke serial monitor.
- **Blynk.virtualWrite(V0, persenCahaya):** Mengirimkan data intensitas cahaya ke aplikasi Blynk pada virtual pin V0.

### 3. SS DASHBOARD



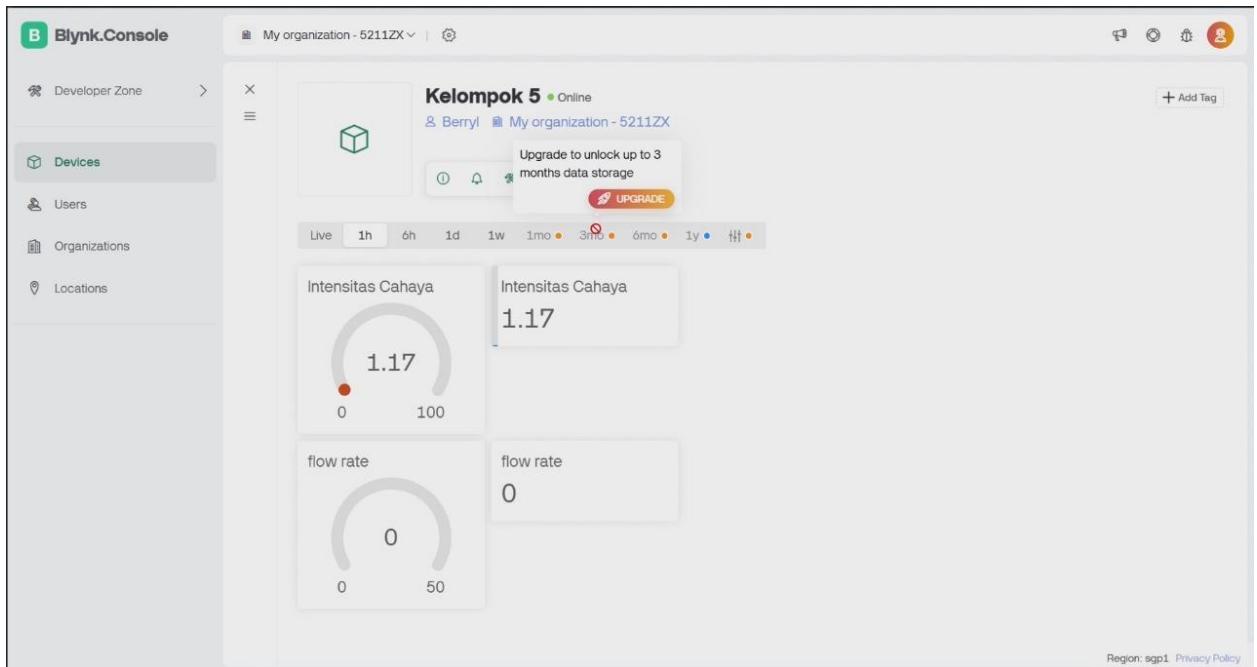
### 4. HASIL PERCOBAAN

- 1 Sendok Garam:



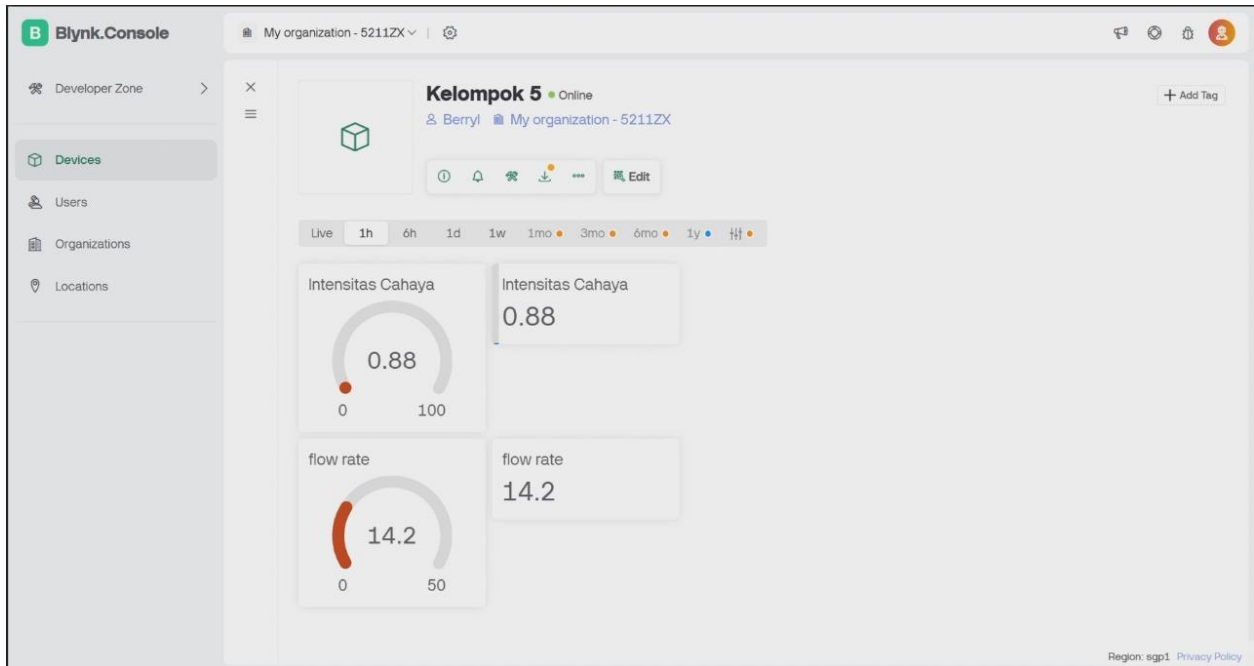


- Berdasarkan gambar di atas, berikut penjelasan tentang dashboard Blynk yang sedang digunakan:
  - Nilai intensitas cahaya yang diperoleh (1.56) menunjukkan bahwa larutan garam mempengaruhi intensitas cahaya yang diterima oleh sensor, mungkin karena perubahan indeks bias atau tingkat kekeruhan air.
- 2 Sendok Garam



- Nilai 1.17 menunjukkan bahwa sensor photodiode mendeteksi sejumlah kecil intensitas cahaya. Ini bisa disebabkan oleh larutan garam yang mungkin mempengaruhi cahaya yang diterima oleh sensor.
- Garam yang dilarutkan dalam air dapat mempengaruhi transparansi larutan. Larutan yang lebih pekat mungkin akan menyerap lebih banyak cahaya, sehingga intensitas yang terukur lebih rendah.

- Menggunakan Pasir/Tanah



- Intensitas Cahaya (Keruhnya Air):
  - Nilai yang terukur untuk intensitas cahaya adalah 0.88.
  - Sensor photodiode digunakan untuk mengukur intensitas cahaya yang melewati air. Nilai yang rendah menunjukkan bahwa air cukup keruh, karena kekeruhan air akan menghambat jalannya cahaya.
  - Grafik atau tampilan menunjukkan intensitas cahaya yang terukur dalam skala 0 hingga 100, dimana 0 adalah paling keruh dan 100 adalah paling jernih.
- Flow Rate (Debit Air):
  - Nilai yang terukur untuk flow rate adalah 14.2.
  - Sensor water flow digunakan untuk mengukur debit air yang mengalir melalui keran air. Flow rate diukur dalam satuan tertentu (misalnya liter per menit).
  - Grafik atau tampilan menunjukkan flow rate yang terukur dalam skala 0 hingga 50, dimana 0 adalah tidak ada aliran dan 50 adalah aliran maksimum.

Dengan menggunakan dashboard ini, pengguna dapat dengan mudah memantau kondisi air dalam media tanah secara terus-menerus, mengidentifikasi perubahan dalam kekeruhan air dan aliran air, serta mengambil tindakan yang diperlukan berdasarkan data yang terukur.

## 5. DOKUMENTASI ALAT

