

KOMUNIKASI DATA SISTEM UPLINK BLYNK

A. Tujuan

1. Memahami konsep dasar Blynk dan cara kerja sistem *uplink* pada ESP32.
2. Menggunakan *website* Blynk untuk mengontrol dan memonitor perangkat yang terhubung dengan ESP32.
3. Menerapkan sistem *uplink* antara ESP32 dan Blynk menggunakan koneksi Wi-Fi untuk pengendalian perangkat.
4. Menganalisis data yang dikirim dan diterima melalui Blynk.

B. Teori Dasar

Dalam komunikasi data, *uplink* adalah proses pengiriman data dari perangkat ke *server* atau *cloud*. *Uplink* dapat diartikan mengirimkan informasi, seperti data sensor atau status perangkat, ke platform pemrosesan atau aplikasi yang dapat diakses pengguna. Blynk adalah platform IoT berbasis *cloud* yang memungkinkan pengguna mengontrol dan memonitor perangkat secara *real-time* melalui aplikasi *mobile* atau *web*. Agar ESP32 dapat terhubung ke Blynk, diperlukan beberapa komponen utama, yaitu:

- Blynk Library untuk menghubungkan perangkat dengan server.
- Authentication Token sebagai kunci akses ke proyek di aplikasi Blynk.
- Wi-Fi SSID & Password agar perangkat dapat terkoneksi ke internet.

Pada jobsheet ini akan dilakukan praktik *monitoring* melalui Serial Monitor dan Blynk, dengan *input* berupa sensor suhu (DHT) dan cahaya (LDR).

C. Alat & Bahan

- a. ESP32
- b. Sensor DHT
- c. Sensor LDR
- d. Breadboard
- e. Kabel jumper
- f. Laptop terinstal Arduino IDE dan library
- g. Kabel USB Type C

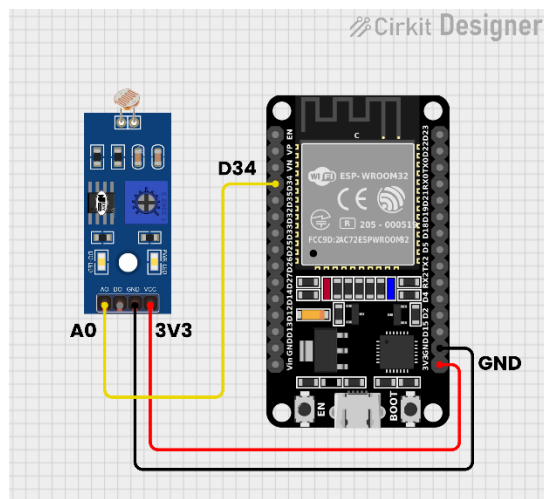
D. Keselamatan Kerja

1. Lakukan, atau laksanakan praktikum di tempat yang bersih dan kering.
2. Menggunakan laptop, atau komputer dengan baik, dan benar.
3. Jangan sentuh langsung bagian kit praktikum yang dialiri listrik.
4. Pastikan *power supply* kit praktikum dalam keadaan mati ketika proses instalasi, atau perakitan kit praktikum.
5. Selalu perhatikan *port* kit praktikum. Jangan sampai tertukar, terbalik, atau salah penempatan dalam instalasi, atau perakitan pada kit praktikum.
6. Gunakan komponen pada kit praktikum sebagaimana mestinya.
7. Jika terjadi *short circuit*, segera matikan *power supply* pada kit praktikum.
8. Mengikuti langkah-langkah praktikum seperti yang tertera pada langkah kerja serta memperhatikan keselamatan kerja.

E. Langkah Kerja

6.1 Monitoring LDR from Serial Monitor

- a. Buat rangkaian seperti gambar berikut. Gunakan pin ADC, karena LDR ini menggunakan *input* analog.



- b. Masukkan kode berikut.

```
#define LDR_PIN 34 // Gunakan pin analog yang sesuai pada ESP32

void setup() {
  Serial.begin(115200); // Memulai komunikasi serial
}

void loop() {
  int ldrValue = analogRead(LDR_PIN); // Membaca nilai dari
  sensor LDR
```

```

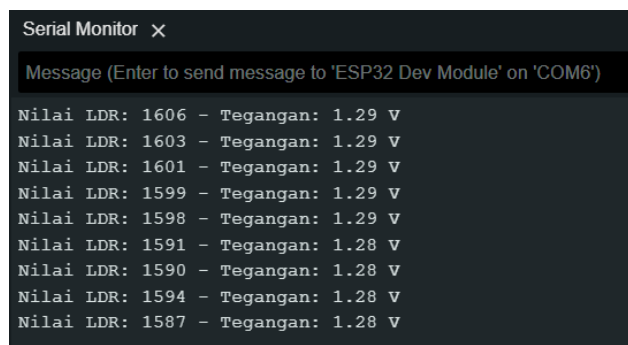
    float voltage = (ldrValue / 4095.0) * 3.3; // Konversi nilai
    ADC ke tegangan

    Serial.print("Nilai LDR: ");
    Serial.print(ldrValue);
    Serial.print(" - Tegangan: ");
    Serial.print(voltage);
    Serial.println(" V");

    delay(1000); // Delay 1 detik sebelum membaca kembali
}

```

- c. Hubungkan ESP32 ke laptop, sebelum Verify atau Compile code pastikan Board ESP32 sudah terhubung di Arduino IDE.
- d. Apabila proses compile selesai dan tidak ada error, selanjutnya upload program, lalu tunggu hingga selesai.
- e. Masuk ke Serial Monitor seperti gambar berikut. Maka tampil data “Nilai LDR” dalam bentuk nilai Analog, kemudian diubah dalam bentuk tegangan melalui persamaan $(ldrValue / 4095.0) * 3.3$; // Konversi nilai ADC ke tegangan.



```

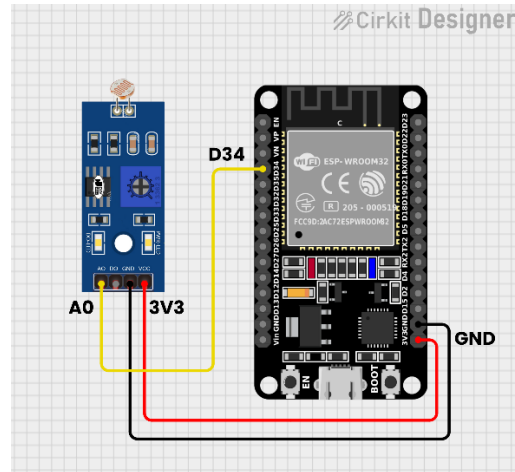
Serial Monitor ✕
Message (Enter to send message to 'ESP32 Dev Module' on 'COM6')

Nilai LDR: 1606 - Tegangan: 1.29 V
Nilai LDR: 1603 - Tegangan: 1.29 V
Nilai LDR: 1601 - Tegangan: 1.29 V
Nilai LDR: 1599 - Tegangan: 1.29 V
Nilai LDR: 1598 - Tegangan: 1.29 V
Nilai LDR: 1591 - Tegangan: 1.28 V
Nilai LDR: 1590 - Tegangan: 1.28 V
Nilai LDR: 1594 - Tegangan: 1.28 V
Nilai LDR: 1587 - Tegangan: 1.28 V

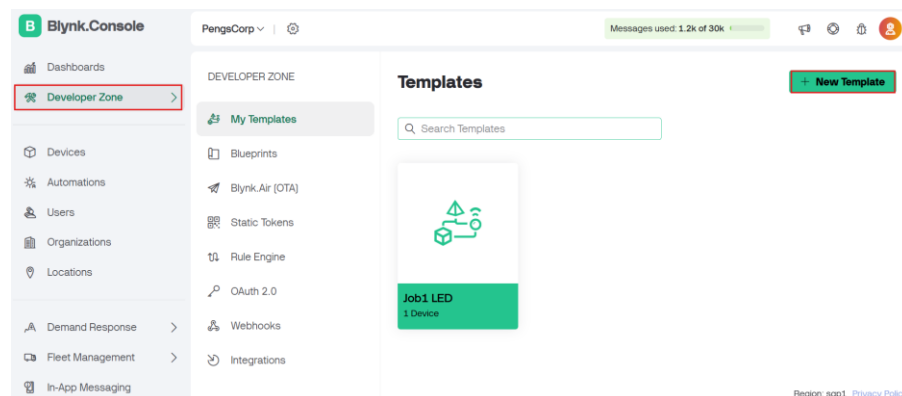
```

6.2 Monitoring LDR from Blynk

- a. Buat rangkaian seperti gambar berikut. Gunakan pin ADC, karena LDR ini menggunakan *input* analog.



- b. Diperlukan konfigurasi terlebih dahulu agar Blynk dan ESP32 bisa saling terkoneksi. Pertama yaitu buat akun Blynk melalui situs <https://blynk.io/>.
- c. Setelah pembuatan akun maka akan diarahkan ke Dashboard utama. Pergi ke menu **Developer Zone** > klik **New Template**.



- d. Isi sesuai dengan kebutuhan, untuk deskripsi opsional.

Create New Template

NAME
LDR to Blynk 12 / 50

HARDWARE
ESP32

CONNECTION TYPE
WiFi

DESCRIPTION
This project for monitoring LDR sensor from Blynk

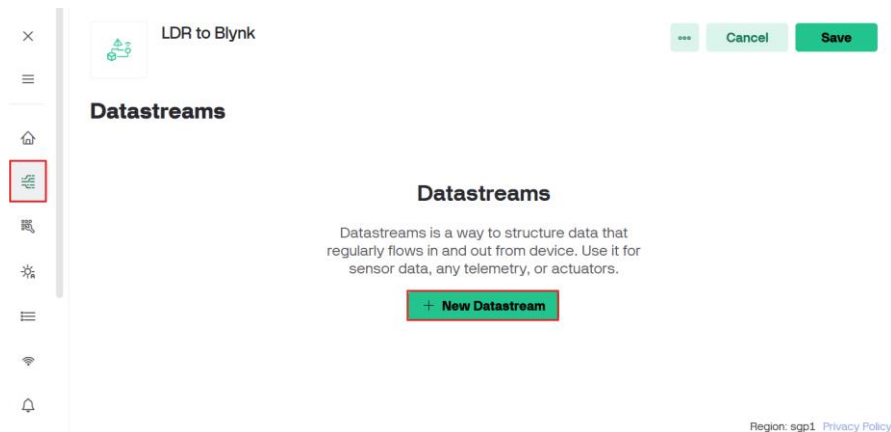
49 / 128

Cancel Done

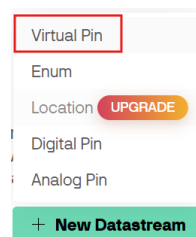
e. Simpan kode tersebut (nantinya) untuk konfigurasi di Arduino IDE.

```
#define BLYNK_TEMPLATE_ID "TMPL6_MYPZj-q"  
#define BLYNK_TEMPLATE_NAME "LDR to Blynk"  
Region: sgp1
```

f. Selanjutnya setting **Datastreams** agar data sensor hingga aktuator dapat masuk atau keluar dari Blynk, dengan klik **New Datastream**.



g. Pilih **Virtual Pin**




h. Isi sesuai kebutuhan dengan ketentuan sebagai berikut:

- **NAME** digunakan untuk memberikan nama Pin.
- **PIN** digunakan sebagai virtual pin pada Blynk.
- **DATA TYPE** sesuai dengan data nilai yang dibaca.
- **UNITS** merupakan satuan data, karena hanya menggunakan bacaan nilai ADC maka dibiarkan **None**.
- Sensor LDR membaca nilai analog maka nilai **MIN** dan **MAX** diubah menjadi 0 – 4095, karena ESP32 menggunakan resolusi 12 bit.

Setelah konfigurasi LDR pin data di atas kemudian klik **Create**.

Virtual Pin Datastream


NAME	LDR_pin		ALIAS	LDR pin		
PIN	V34		DATA TYPE	Integer		
UNITS	None					
MIN	0		MAX	4095		DEFAULT VALUE
				0		
<input type="checkbox"/> Enable history data						
+ ADVANCED SETTINGS						

[Cancel](#)[Create](#)

- i. Selanjutnya konfigurasi pin Intensitas Cahaya dengan luaran tegangan.


Virtual Pin Datastream

General [Expose to Automations](#)

NAME	Intensitas Cahaya		ALIAS	Intensitas Cahaya		
PIN	V1		DATA TYPE	Integer		
UNITS	None					
MIN	0		MAX	3.3		DEFAULT VALUE
				0		
<input type="checkbox"/> Enable history data						

[Cancel](#)[Create](#)

- j. Setelah itu klik **Save**.





 LDR to Blynk

[Cancel](#) [Save And Apply](#)

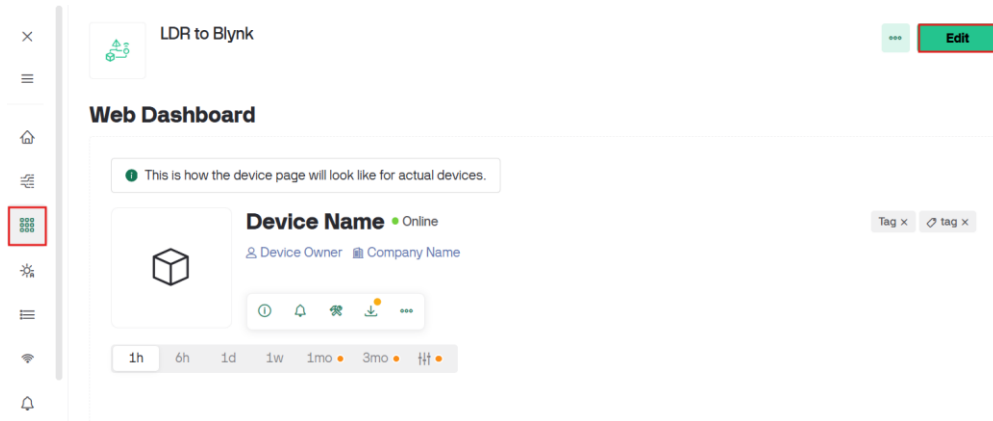
Datastreams

[+ New Datastream](#)

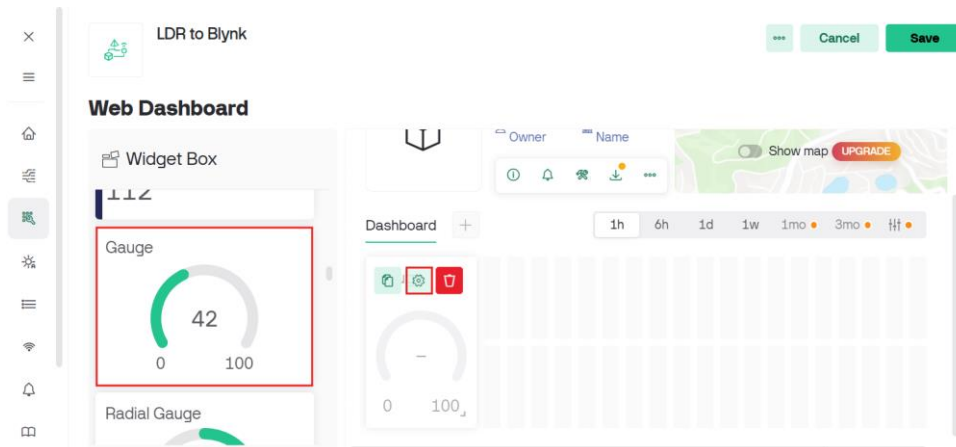
2 Datastreams

<input type="checkbox"/>	Id	Name	Pin	Color	Data Type	Units	Is Raw	Min	Max	Δ	Actions
	1	LDR_pin	V34		Integer		false	0	4095	-	
	2	Intensitas Cahaya	V1		Integer		false	0	3.3	-	

k. Selanjutnya konfigurasi **Web Dashboard** dengan klik menu **Edit**.



l. Pilih **Widget** sesuai dengan kebutuhan, lalu tarik ke sebelah kanan seperti gambar berikut. Jika sudah klik ikon **pengaturan** pada widget.



m. Masukkan **TITLE** sesuai kebutuhan, selanjutnya pilih **Datastreams** yang telah dibuat yaitu **LDR_pin(V34)**. Selanjutnya klik **Save**.

Gauge Settings

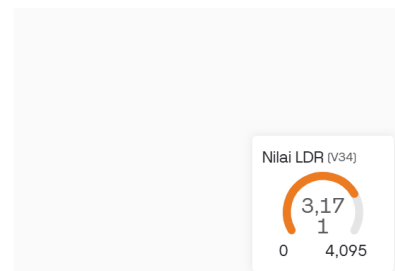
TITLE (OPTIONAL)

Datastream

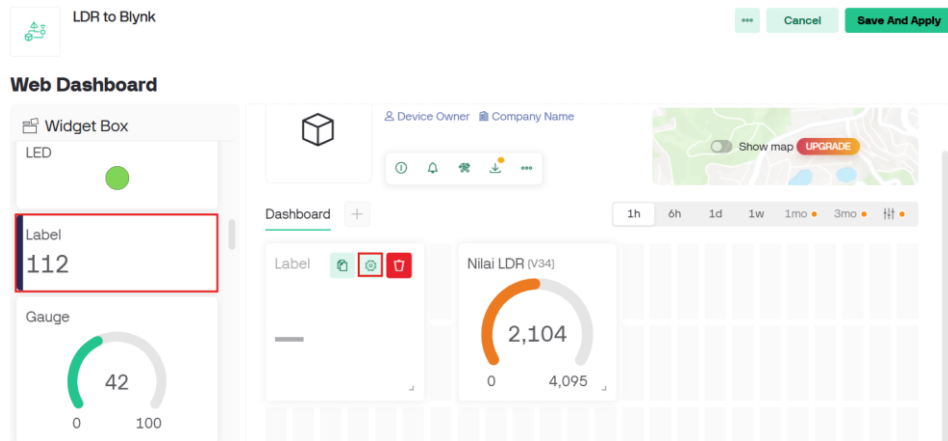
☐ Override Datastream's Min/Max fields

LEVEL COLOR
☐ Change color based on value

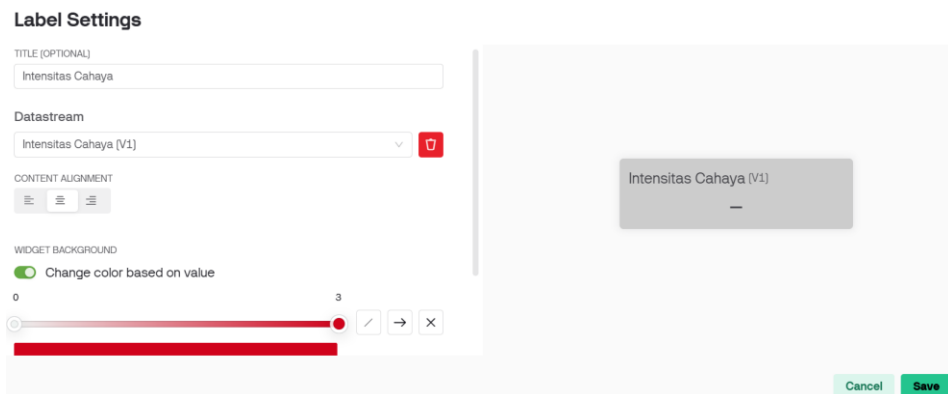
☐



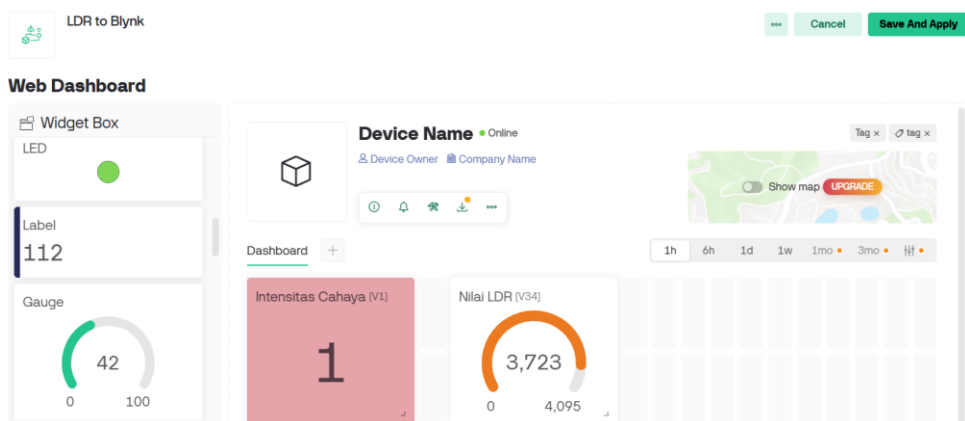
n. Tambahkan lagi Widget untuk Intensitas Cahaya.



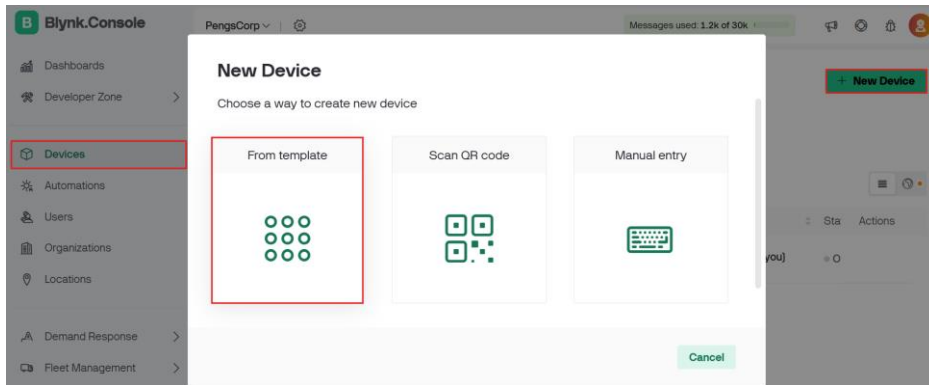
- o. Isi kolom Datastream dengan Intensitas Cahaya (V1) yang telah dibuat tadi.



- p. Apabila telah selesai konfigurasi **Web Dashboard**, klik **Save**.



- q. Selanjutnya pergi ke menu **Devices** untuk menambahkan perangkat virtual pada Blynk. Klik **New Device** > lalu pilih **From template**.



- r. Pada kolom **TEMPLATE**, pilih template yang telah dibuat. Lalu masukkan **DEVICE NAME** sesuai kebutuhan, dan klik **Create**.

New Device

Create new device by filling in the form below

TEMPLATE

LDR to Blynk

DEVICE NAME

LDR to Blynk

12 / 50

Cancel

Create

- s. Simpan kode berikut.

```
#define BLYNK_TEMPLATE_ID "TMPL6_MypZj-q"
#define BLYNK_TEMPLATE_NAME "LDR to Blynk"
#define BLYNK_AUTH_TOKEN "1SY9-D1bQiV5l-c2NWsvj0dRLLqBKZt7"
```

- t. Konfigurasi Blynk telah selesai, selanjutnya konfigurasi program pada Arduino IDE. Masukkan kode berikut, beberapa hal yang perlu disesuaikan kembali:

- Blynk template IDE
- Blynk template name
- Blynk device name
- Token autentikasi Blynk
- SSID WiFi
- Password WiFi, dan
- Virtual Pin

```
#define BLYNK_TEMPLATE_ID "TMPL6_MypZj-q"
#define BLYNK_TEMPLATE_NAME "LDR to Blynk"
#include <WiFi.h>
#include <BlynkSimpleEsp32.h>
```

```
// Konfigurasi WiFi
```

```
char ssid[] = "realme"; // Sesuaikan dengan jaringan anda
```

```

char pass[] = "12345678";

// Token Blynk
char auth[] = "1SY9_D1bQiV5l-c2NWsvj0dRlLqBKZt7"; // Ganti
dengan token Anda

// Pin LDR
#define LDR_PIN 34

void setup() {
  Serial.begin(115200); // Baud rate Serial Monitor
  Blynk.begin(auth, ssid, pass); // Inisialisasi Blynk
}

void loop() {
  Blynk.run(); // Jalankan Blynk

  int ldrValue = analogRead(LDR_PIN);
  float voltage = (ldrValue * 3.3) / 4095.0; // Konversi ADC ke
nilai tegangan

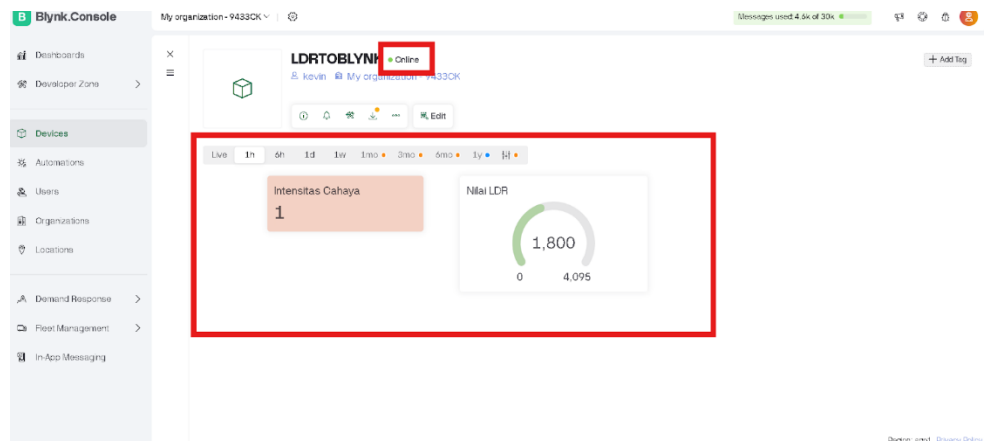
  // Kirim data ke Blynk
  Blynk.virtualWrite(V34, ldrValue); // Ubah virtual pin sesuai
konfigurasi
  Blynk.virtualWrite(V1, voltage);

  Serial.print("Nilai LDR: ");
  Serial.print(ldrValue);
  Serial.print(" | Tegangan: ");
  Serial.println(voltage);

  delay(1000);
}

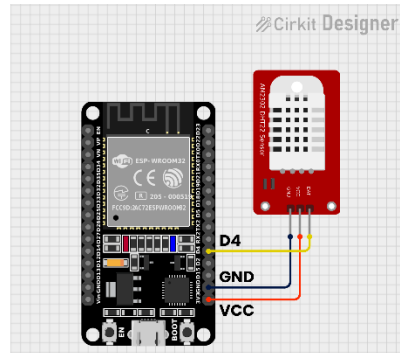
```

- u. Hubungkan ESP32 ke laptop, sebelum Verify atau Compile code pastikan Board ESP32 sudah terhubung di Arduino IDE, lalu upload program.
- v. Periksa hasil melalui Blynk, pastikan device statu Online, lihat nilai Intensitas Cahaya dan nilai LDR di Widget.



6.3 Monitoring DHT from Serial Monitor

- a. Buat rangkaian seperti gambar berikut. Gunakan pin Digital, karena DHT menggunakan digital input.



- f. Masukkan kode berikut.

```
// import library sensor dht
#include <DHT.h>

// Konfigurasi pin dan tipe sensor DHT
#define DHTPIN 4      // Pin data DHT22 ke GPIO4 ESP32
#define DHTTYPE DHT11 // Tipe sensor DHT22

DHT dht(DHTPIN, DHTTYPE); // Inisialisasi objek DHT

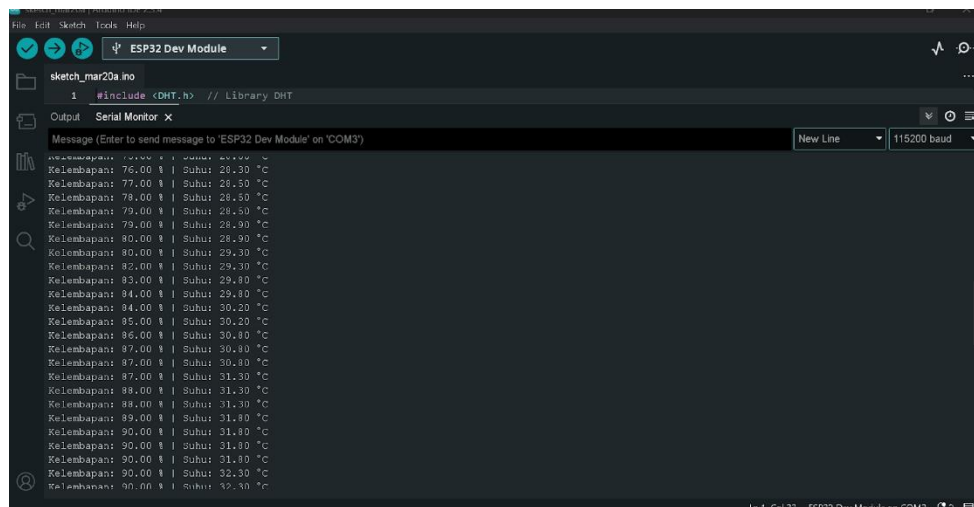
void setup() {
  Serial.begin(115200);
  Serial.println("Memulai Sensor DHT22...");
  dht.begin(); // Memulai sensor DHT22
}

void loop() {
  float temperature = dht.readTemperature(); // Membaca suhu
  dalam °C
  float humidity = dht.readHumidity();       // Membaca
  kelembaban dalam %

  // Cek apakah pembacaan valid
  if (isnan(temperature) || isnan(humidity)) {
    Serial.println("Gagal membaca dari DHT22!");
  } else {
    // Tampilkan hasil ke Serial Monitor
    Serial.print("Suhu: ");
    Serial.print(temperature);
    Serial.print("°C | Kelembaban: ");
    Serial.print(humidity);
    Serial.println("%");
  }

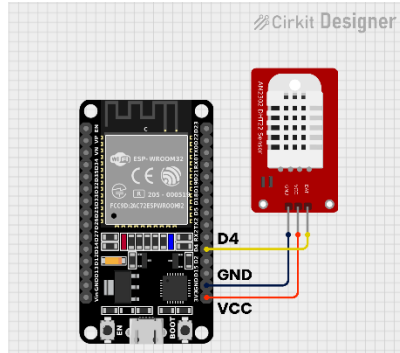
  delay(2000); // Tunggu 2 detik sebelum membaca kembali
}
```

- g. Arduino IDE memerlukan library sensor DHT22 agar mempermudah pembacaan sensor DHT22 tanpa harus menulis kode komunikasi digital secara manual.
- h. Hubungkan ESP32 ke laptop, sebelum Verify atau Compile code pastikan Board ESP32 sudah terhubung di Arduino IDE.
- i. Apabila proses compile selesai dan tidak ada error, selanjutnya upload program, lalu tunggu hingga selesai.
- j. Masuk ke Serial Monitor seperti gambar berikut.

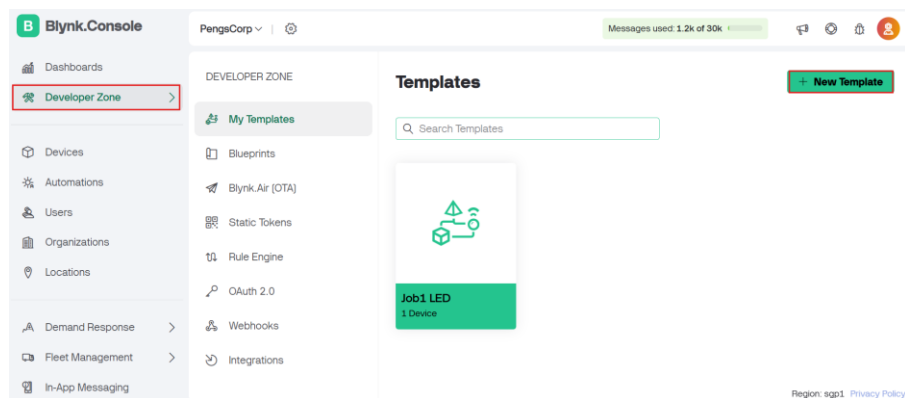


6.4 Monitoring DHT from Blynk

- Buat rangkaian seperti gambar berikut. Gunakan pin Digital, karena DHT menggunakan digital input.



- Selanjutnya diperlukan konfigurasi Blynk agar bisa saling terkoneksi dengan ESP32.
- Pergi ke menu **Developer Zone** > klik **New Template**.



- Isi sesuai dengan kebutuhan, untuk deskripsi opsional.

Create New Template

NAME
DHT BLYNK 9 / 50

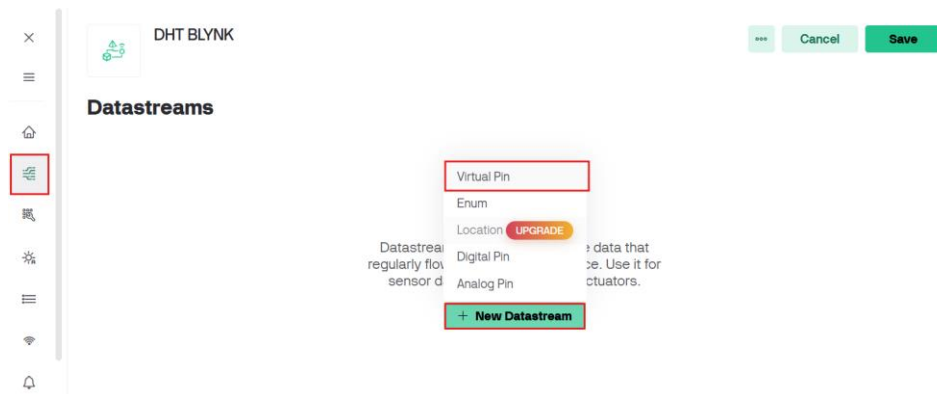
HARDWARE ESP32 CONNECTION TYPE WiFi

DESCRIPTION
Description 0 / 128

Cancel

Done

- e. Selanjutnya, masuk ke menu **Datastreams** > klik **New Datastream** > pilih **Virtual Pin**.



- f. Masukkan Datastream Suhu sesuai dengan kebutuhan, untuk nilai **MIN** dan **MAX** berupa suhu dari 0 - 50° C.

Virtual Pin Datastream

General Expose to Automations

NAME	ALIAS	
<input type="text" value="Suhu"/>	<input type="text" value="Suhu"/>	
PIN	DATA TYPE	
<input type="text" value="V1"/>	<input type="text" value="Integer"/>	
UNITS		
<input type="text" value="None"/>		
MIN	MAX	DEFAULT VALUE
<input type="text" value="0"/>	<input type="text" value="50"/>	<input type="text" value="0"/>

Cancel Create

- g. Kemudian tambahkan Datastream lagi untuk Kelembaban, untuk nilai **MIN** dan **MAX** yaitu dari 0 – 100%.

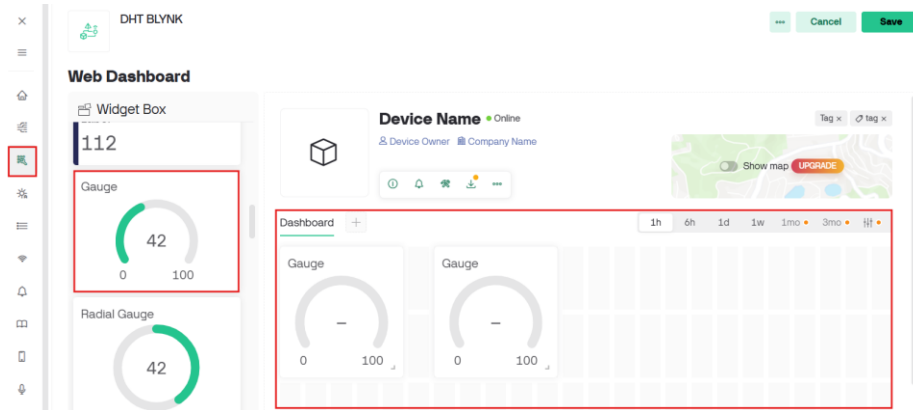
Virtual Pin Datastream

General Expose to Automations

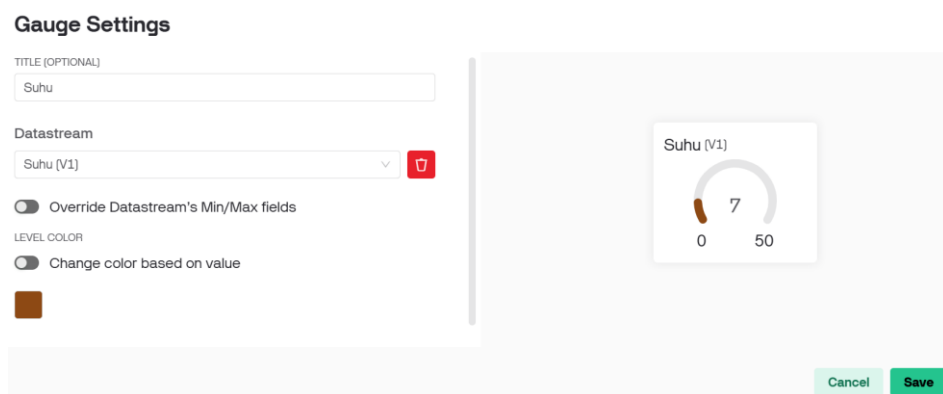
NAME	ALIAS	
<input type="text" value="Kelembaban"/>	<input type="text" value="Kelembaban"/>	
PIN	DATA TYPE	
<input type="text" value="V2"/>	<input type="text" value="Integer"/>	
UNITS		
<input type="text" value="None"/>		
MIN	MAX	DEFAULT VALUE
<input type="text" value="0"/>	<input type="text" value="100"/>	<input type="text" value="0"/>

Cancel Create

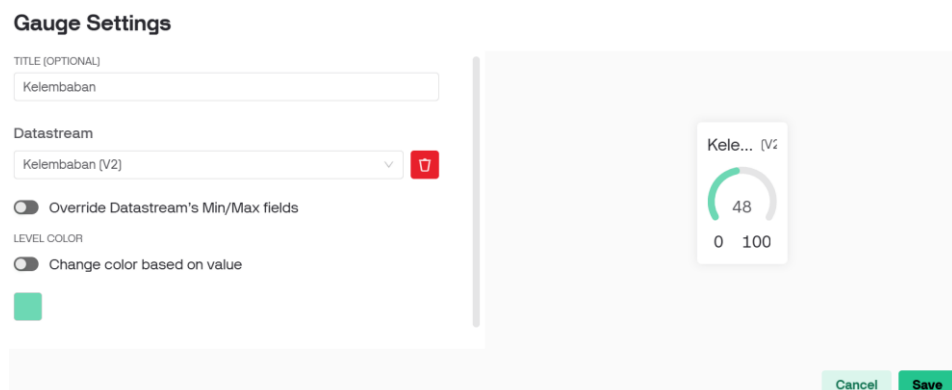
- h. Selanjutnya konfigurasi menu **Web Dashboard**, drag and drop **Widget** ke bagian **Dashboard** sebelah kanan sebanyak dua buah untuk Suhu dan Kelembaban.



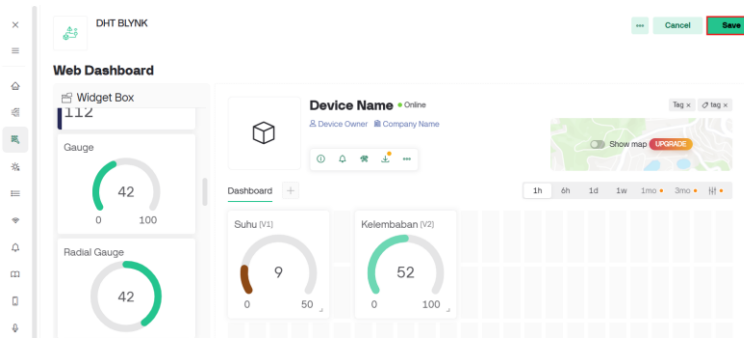
- i. Konfigurasi Gauge 1 dengan sumber Datastream **Suhu (V1)**, lalu klik **Save**.



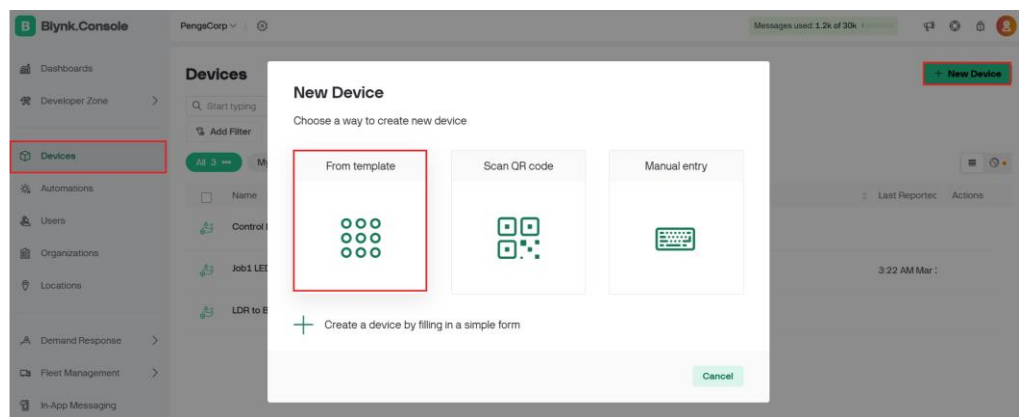
- j. Selanjutnya konfigurasi Gauge 2 dengan sumber Datastream **Kelembaban (V2)**, lalu klik **Save**.



k. Jika selesai konfigurasi **Web Dashboard** klik **Save**.



l. Selanjutnya masuk ke menu **Devices** untuk mengatur perangkat virtual.
Klik **New Device** > pilih **From template**.



m. Pada kolom **TEMPLATE** pilih template yang telah dibuat pada tahap awal tadi yaitu **DHT BLYNK**, jika sudah klik **Create**.

New Device

Create new device by filling in the form below

TEMPLATE

DHT BLYNK

DEVICE NAME

DHT BLYNK 9 / 50

Cancel Create

n. Simpan kode tersebut untuk konfigurasi di Arduino IDE.

New Device Created!



```
#define BLYNK_TEMPLATE_ID "TMPL6ubmFc1C1"
#define BLYNK_TEMPLATE_NAME "DHT BLYNK"
#define BLYNK_AUTH_TOKEN "qDD37SGObXCRpfPMwRi7B3eFMBw6FMDe"
```


- o. Tahap selanjutnya, konfigurasi program melalui Arduino IDE, masukkan kode berikut.

```
#define BLYNK_TEMPLATE_ID "TMPL6ubmFc1C1"
#define BLYNK_TEMPLATE_NAME "DHT BLYNK"
#define BLYNK_AUTH_TOKEN "qDD37SGObXCRpfPMwRi7B3eFMBw6FMDe"

#include <WiFi.h>
#include <BlynkSimpleEsp32.h>
#include <DHT.h> // DHT_kxn

#define DHTPIN 4 // definisi pin sensor DHT
#define DHTTYPE DHT11

DHT dht(DHTPIN, DHTTYPE); // Konstruktor DHT dengan pin dan tipe sensor

char ssid[] = "fxx";
char pass[] = "11111111";

BlynkTimer timer;

void setup() {
  Serial.begin(115200);
  dht.begin(); // Inisialisasi sensor
  Blynk.begin(BLYNK_AUTH_TOKEN, ssid, pass);
  timer.setInterval(2000L, sendDataToBlynk);
}

void loop() {
  Blynk.run();
  timer.run();
}

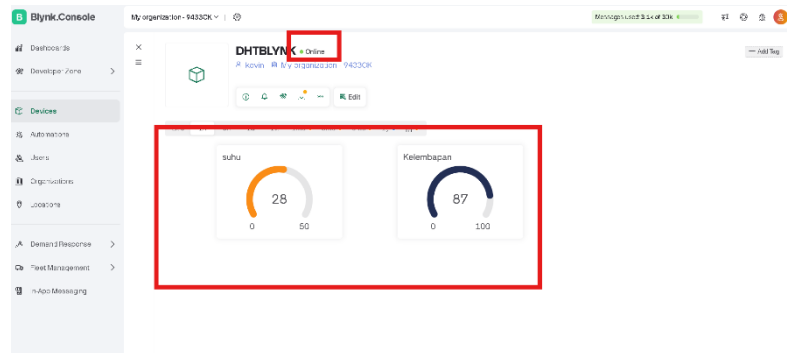
void sendDataToBlynk() {
  float temperature = dht.readTemperature(); // <- benar
  float humidity = dht.readHumidity();       // <- benar

  if (isnan(temperature) || isnan(humidity)) {
    Serial.println("Gagal membaca sensor!");
    return;
  }

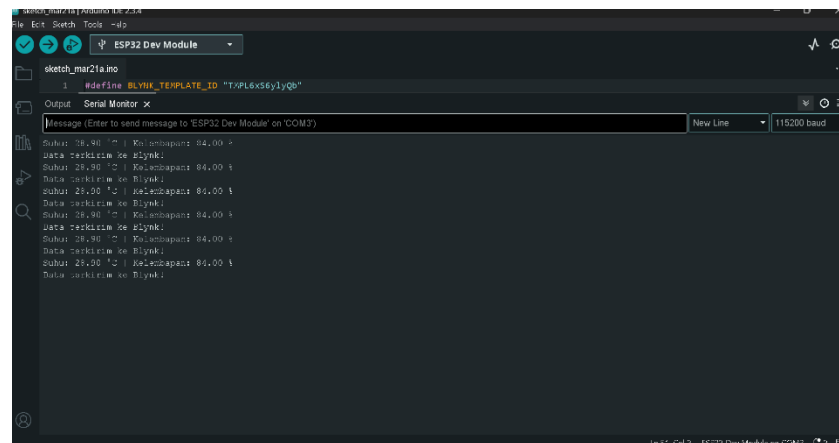
  Blynk.virtualWrite(V0, temperature); // ganti sesuai pin virtual di Blynk
  Blynk.virtualWrite(V1, humidity); // ganti sesuai pin virtual di Blynk
  Serial.println("Data terkirim ke Blynk!");
}
```

- p. Hubungkan ESP32 ke laptop, sebelum Verify atau Compile code pastikan Board ESP32 sudah terhubung di Arduino IDE.

- q. Apabila proses compile selesai dan tidak ada error, selanjutnya upload program, lalu tunggu hingga selesai.
- r. Untuk melihat hasil, periksa di website Blynk. Jika sudah terhubung maka status akan menjadi Online.



- s. Bisa juga memeriksa hasil melalui Serial Monitor, dengan mengatur baud rate 115200.



F. Soal

1. Tampilkan output berdasarkan dari instruksi Langkah Kerja! Tambahkan analisis dan prinsip kerjanya!
2. Ubah widget pada monitoring sensor DHT atau LDR sesuai keinginan Anda, gunakan resource pada Blynk Widget Box!
3. Dokumentasikan hasil pengerjaan pada lembar Kerja Praktikum. Sertakan video hasil praktikum dan source code-nya!
4. Upload lembar kerja Laporan Praktikum (dalam format PDF) dan video dokumentasi di Google Drive.

<https://drive.google.com/drive/folders/1QWANpOpyaCtYx6A11s2VY5WKJCsV4Sbj?usp=sharing>

5. Ketentuan pengumpulan drive sebagai berikut:
 - Gunakan akun students UNNES
 - Buat folder dengan format “NIM_Nama Lengkap”
 - Upload lembar kerja Laporan Praktikum dan dokumentasi video