

BAB 7

KOMUNIKASI DATA SISTEM DOWNLINK BLYNK

A. Tujuan

1. Memahami konsep *downlink* dalam IoT, yaitu proses penerimaan data atau perintah dari server ke perangkat ESP32 melalui platform Blynk.
2. Menerapkan sistem *downlink* dengan mengonfigurasi ESP32 agar dapat menerima dan menjalankan perintah dari aplikasi Blynk.
3. Menggunakan fungsi dan fitur Blynk, seperti Blynk.virtualRead() atau Blynk.syncVirtual(), untuk mengontrol perangkat IoT secara jarak jauh.
4. Menganalisis efektivitas sistem *downlink* dalam IoT dengan menguji respons perangkat terhadap perintah yang dikirimkan dari Blynk

B. Teori Dasar

Dalam komunikasi data, *downlink* adalah proses penerimaan data atau perintah dari *server* ke perangkat. *Downlink* memungkinkan server atau aplikasi mengirimkan instruksi ke perangkat yang terhubung, seperti ESP32 untuk mengontrol aktuator atau mengubah parameter sistem secara jarak jauh. Untuk mengimplementasikan sistem downlink di Blynk, diperlukan:

- Koneksi Internet & Server Blynk – Perangkat ESP32 harus terhubung ke server Blynk menggunakan Wi-Fi.
- Authentication Token – Digunakan untuk menghubungkan ESP32 dengan proyek di aplikasi Blynk.
- Fungsi Downlink dalam Blynk:
 - BLYNK_WRITE(Vx) – Digunakan untuk membaca nilai dari Virtual Pin yang dikontrol melalui aplikasi Blynk.
 - Blynk.syncVirtual(Vx) – Digunakan untuk menyinkronkan status terakhir Virtual Pin saat perangkat terhubung kembali.

C. Alat & Bahan

- a. ESP32
- b. LED atau Buzzer
- c. Relay

- d. Breadboard
- e. Kabel jumper
- f. Laptop terinstal Arduino IDE dan library
- g. Kabel USB Type C

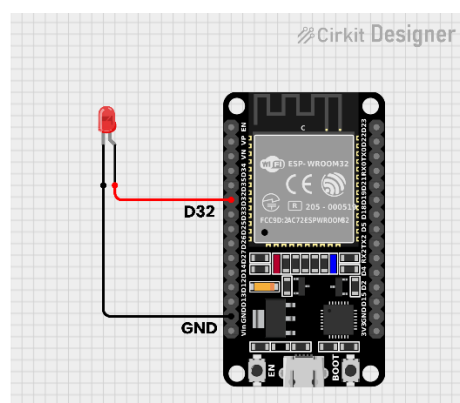
D. Keselamatan Kerja

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

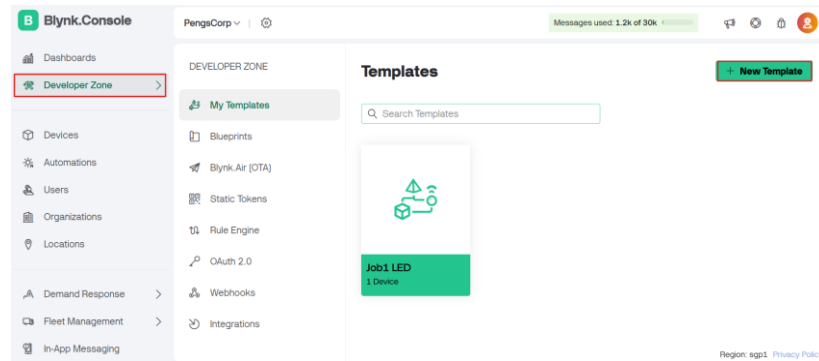
E. Langkah Kerja

6.1 LED Control from Blynk

- a. Buat rangkaian seperti gambar berikut.



- b. Konfigurasi Blynk terlebih dahulu untuk membuat *virtual* platform.
- c. Masuk ke menu **Developer Zone** lalu pilih **New Template**.



d. Isi sesuai dengan kebutuhan, jika sudah klik Done

Create New Template

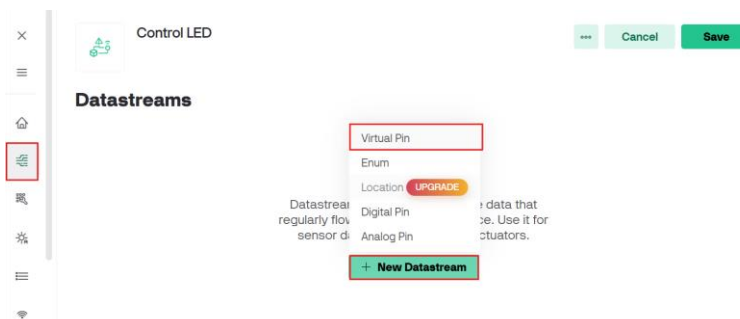
NAME
Control LED 11 / 50

HARDWARE ESP32 CONNECTION TYPE WIFI

DESCRIPTION
Description 0 / 128

Cancel Done

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



f. Masukkan sesuai dengan kebutuhan, karena LED bekerja secara digital yaitu hanya 0 dan 1. Maka *setting* di kolom MIN (0) dan MAX (1), berbeda dengan aktuator analog

Virtual Pin Datastream

General Expose to Automations

NAME LED 1 ALIAS LED 1

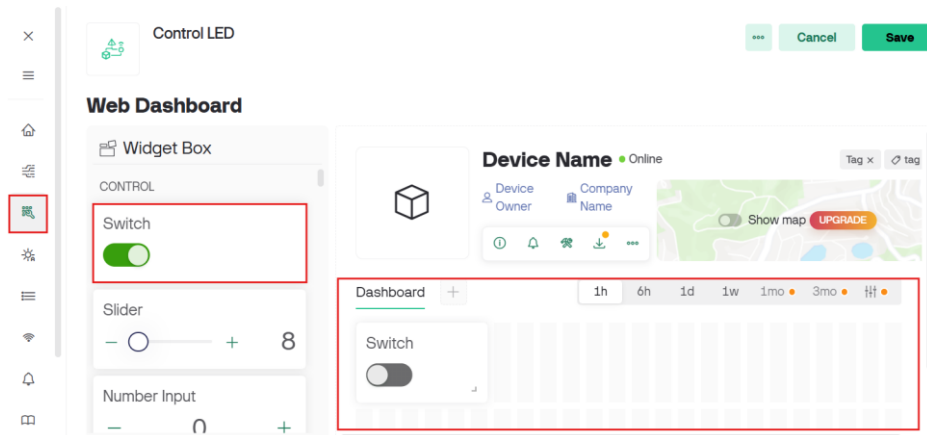
PIN V0 DATA TYPE Integer

UNITS None

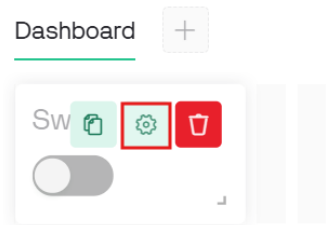
MIN 0 MAX 1 DEFAULT VALUE 0

Cancel Create

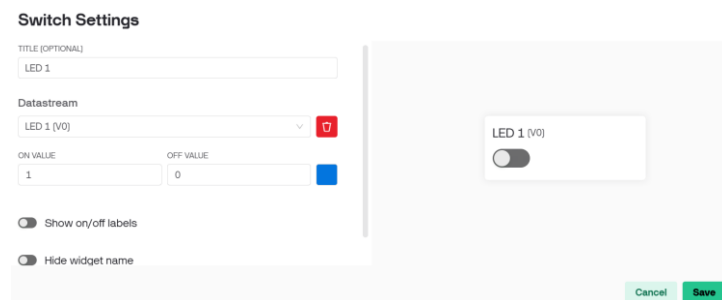
- g. Selanjutnya masuk ke menu **Dashboard** untuk membuat panel kontrol.
Pilih **Switch** > *drag and drop* ke area **Dashboard** sebelah kanan.



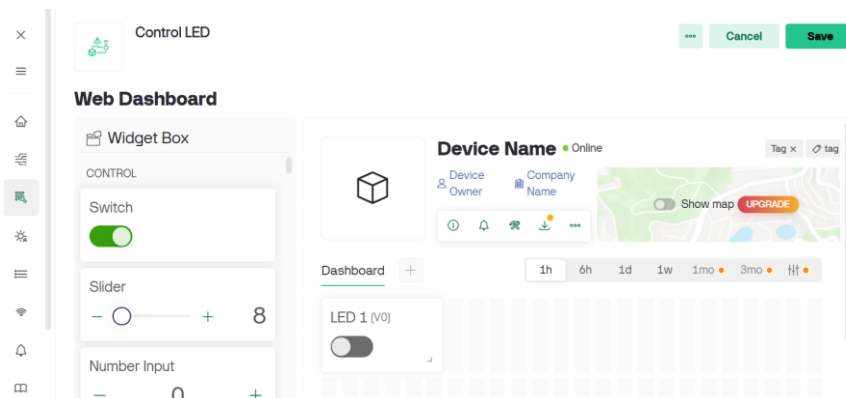
- h. Konfigurasi tombol **Switch** dengan klik icon pengaturan.



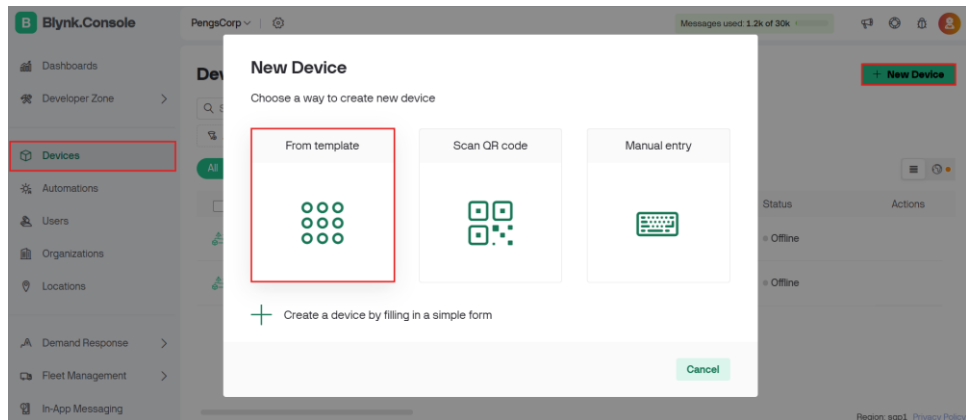
- i. Konfigurasi sesuai dengan kebutuhan, pada kolom **Datastream** pilih datastream yang telah dikonfigurasi tadi. Jika sudah klik **Save**.



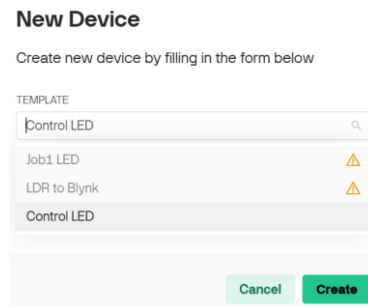
- j. Web Dashboard selesai dikonfigurasi. Klik Save sebelah kanan atas.



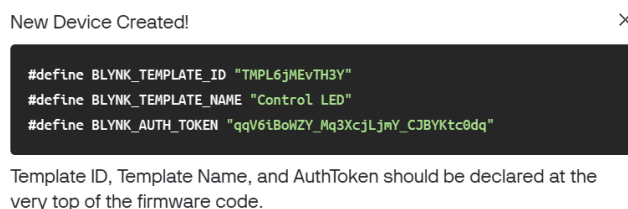
- k. Selanjutnya masuk ke menu Devices untuk mengatur perangkat *virtual*.
Klik **New Device** > pilih **From template**.



- l. Pilih Template yang telah dikonfigurasi, jika sudah klik **Create**.



- m. Sehingga akan muncul kode, simpan kode tersebut untuk dikonfigurasi melalui Arduino IDE.



- n. Selanjutnya konfigurasi program di Arduino IDE. Masukkan kode berikut.

```
#define BLYNK_TEMPLATE_ID "TMPL60vDZ2M_G" // Sesuaikan  
Template ID  
#define BLYNK_TEMPLATE_NAME "job2Led" // Sesuaikan Nama  
template  
#define BLYNK_AUTH_TOKEN "OZLBEnmShKMZD_8AxQV2LfYNpxrIE5No" //  
Sesuaikan Auth Token  
  
#include <WiFi.h>  
#include <BlynkSimpleEsp32.h>  
  
// Konfigurasi WiFi (sesuaikan dengan wifi)  
char ssid[] = "realme"; //  
char pass[] = "12345678"; //  
  
// Pin LED di ESP32 (sesuaikan dengan pin esp)
```

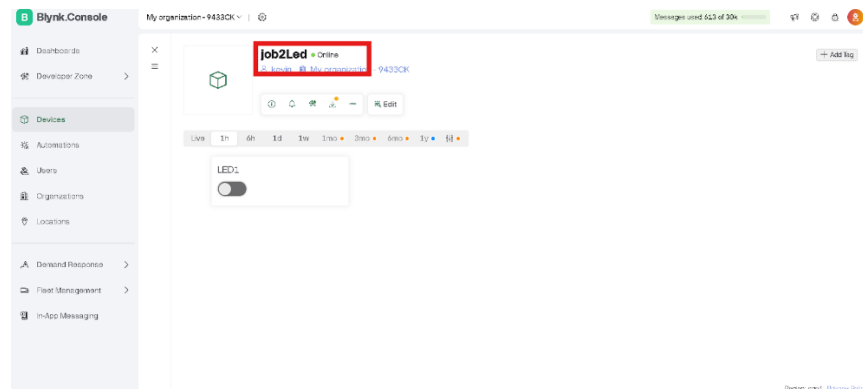
```
#define LED_PIN 32

// Fungsi untuk mengontrol LED via Virtual Pin V0 di Blynk
BLYNK_WRITE(V0) {
  int status = param.asInt(); // Baca nilai dari tombol (0 atau 1)
  digitalWrite(LED_PIN, status); // Aktifkan/matifkan LED
}

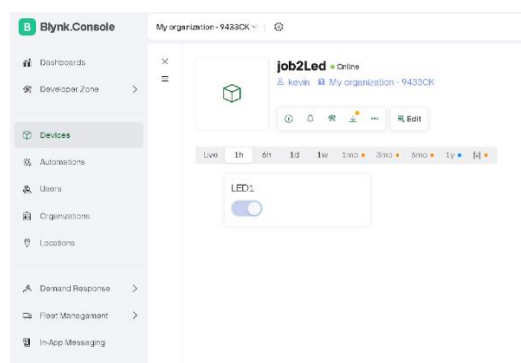
void setup() {
  Serial.begin(115200);
  pinMode(LED_PIN, OUTPUT); // Set pin sebagai output
  digitalWrite(LED_PIN, LOW); // Pastikan LED mati saat boot
  Blynk.begin(BLYNK_AUTH_TOKEN, ssid, pass); // Inisialisasi Blynk
}

void loop() {
  Blynk.run(); // Jalankan proses Blynk
}
```

- o. Hubungkan ESP32 ke laptop, sebelum Verify atau Compile code pastikan Board ESP32 sudah terhubung di Arduino IDE, selanjutnya Upload.
- p. Setelah berhasil Upload, masuk ke Blynk apabila sudah tersambung maka status akan berubah menjadi **Online**.

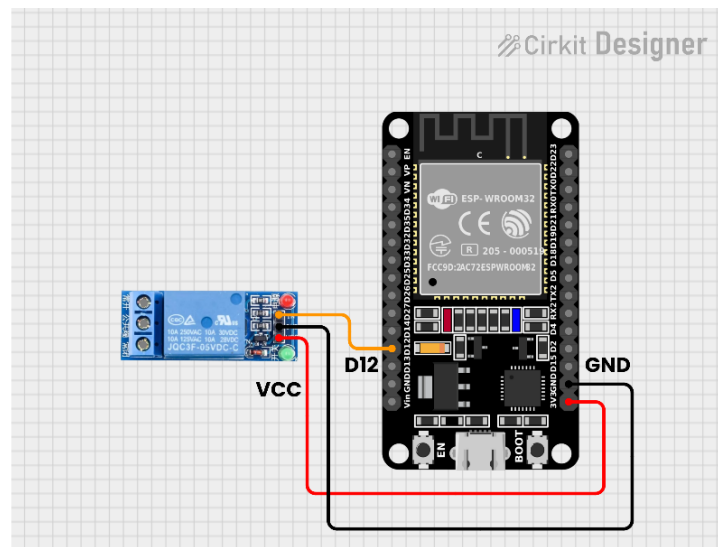


- q. Lakukan percobaan untuk mengaktifkan atau mematikan LED melalui Blynk dengan klik pada tombol Switch.



6.2 Relay Control from Serial Monitor

a. Buat rangkain seperti gambar berikut.



b. Masukkan kode berikut di Arduino IDE.

```
// Pin Relay
#define RELAY_PIN 4 // Sesuaikan dengan Pin yang digunakan

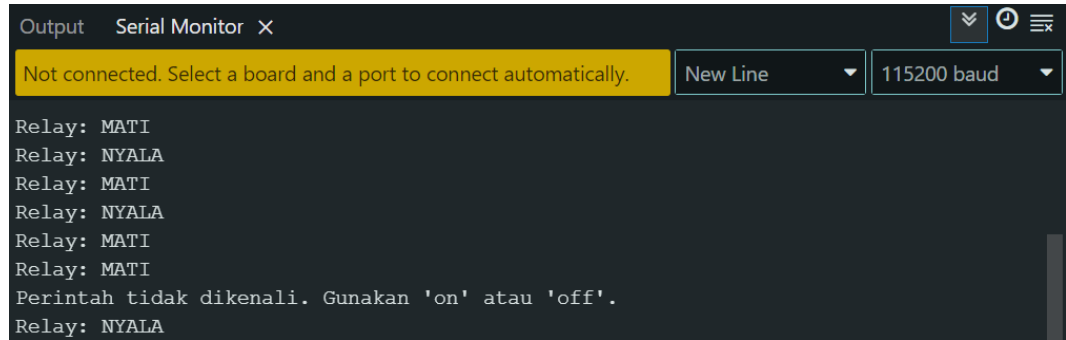
void setup() {
  Serial.begin(115200);
  pinMode(RELAY_PIN, OUTPUT);
  digitalWrite(RELAY_PIN, LOW); // Relay mati saat boot
  Serial.println("Ketik 'on' untuk menyalakan relay, atau 'off'
    untuk mematikan relay.");
}

void loop() {
  if (Serial.available()) {
    String input = Serial.readStringUntil('\n');
    input.trim(); // Hapus spasi dan karakter newline

    if (input.equalsIgnoreCase("on")) {
      digitalWrite(RELAY_PIN, HIGH);
      Serial.println("Relay: NYALA");
    } else if (input.equalsIgnoreCase("off")) {
      digitalWrite(RELAY_PIN, LOW);
      Serial.println("Relay: MATI");
    } else {
      Serial.println("Perintah tidak dikenali. Gunakan 'on' atau
        'off'.");
    }
  }
}
```

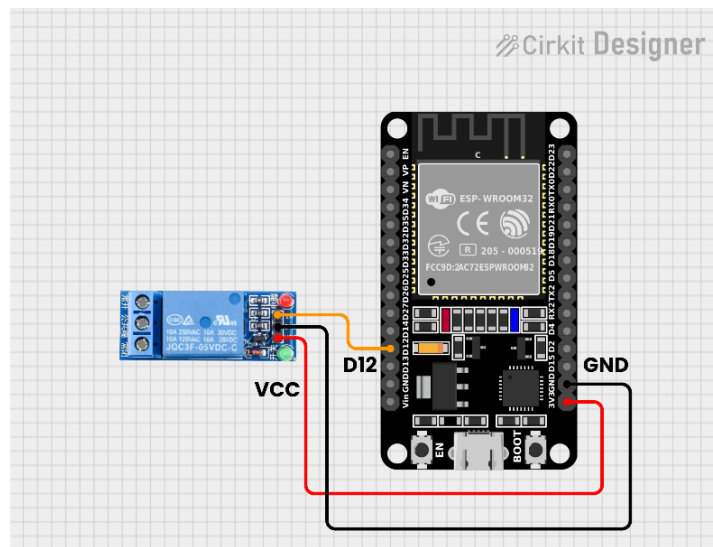
c. Hubungkan ESP32 ke laptop, sebelum Verify atau Compile code pastikan Board ESP32 sudah terhubung di Arduino IDE.

- d. Apabila proses compile selesai dan tidak ada error, selanjutnya upload program, lalu tunggu hingga selesai.
- e. Masuk ke Serial Monitor, lalu ubah baud rate sesuai yang telah diprogram.
- f. Ketik perintah “ON” atau “OFF” seperti yang telah dikonfigurasi, untuk menyalakan atau mematikan LED. Setelah mengetik perintah klik enter.



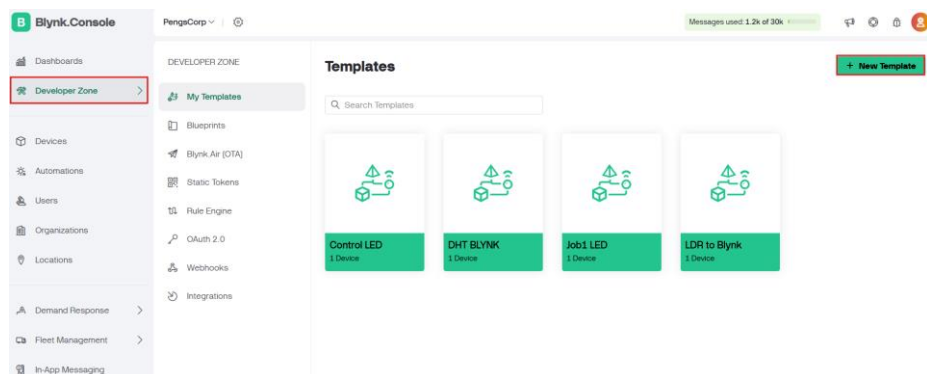
6.3 Relay Control from Blynk

a. Buat rangkaian seperti gambar berikut.



b. Konfigurasi Blynk terlebih dahulu untuk membuat *virtual* platform.

c. Pergi ke menu **Developer Zone** > klik **New Template**.



d. Isi sesuai dengan kebutuhan, jika sudah klik **Done**.

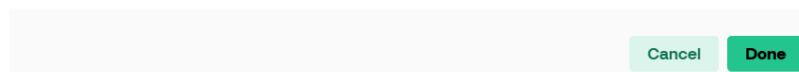
Create New Template

NAME
Blynk Relay 11 / 50

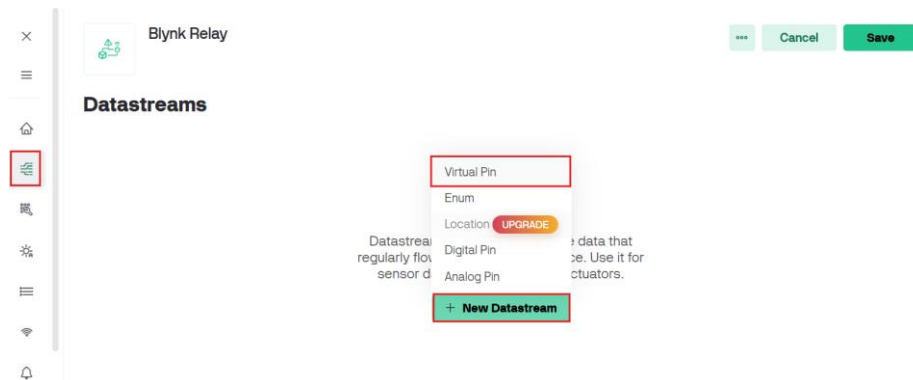
HARDWARE
ESP32

CONNECTION TYPE
WIFI

DESCRIPTION
Description 0 / 128



- e. Kemudian konfigurasi di menu **Datastreams**, klik **New Datastream** > lalu pilih **Virtual Pin**.



- f. Isi sesuai dengan kebutuhan, karena Relay bekerja secara digital maka isi nilai **MIN (0)** dan **MAX (1)**.

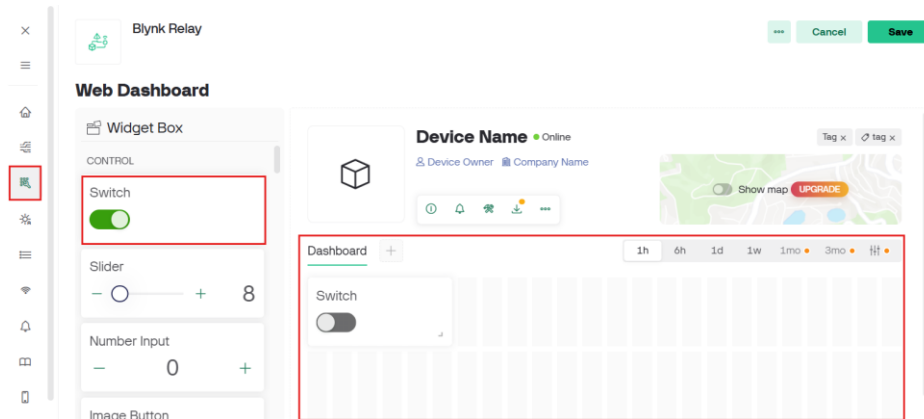
Virtual Pin Datastream

General Expose to Automations

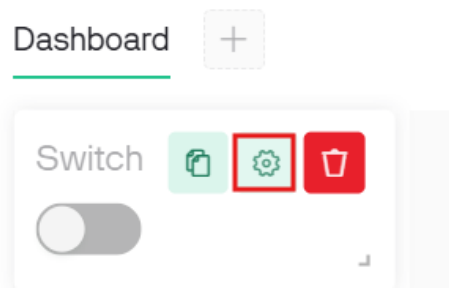
NAME	Relay	ALIAS	Relay
PIN	V1	DATA TYPE	Integer
UNITS	None		
MIN	0	MAX	1
		DEFAULT VALUE	0

Cancel Create

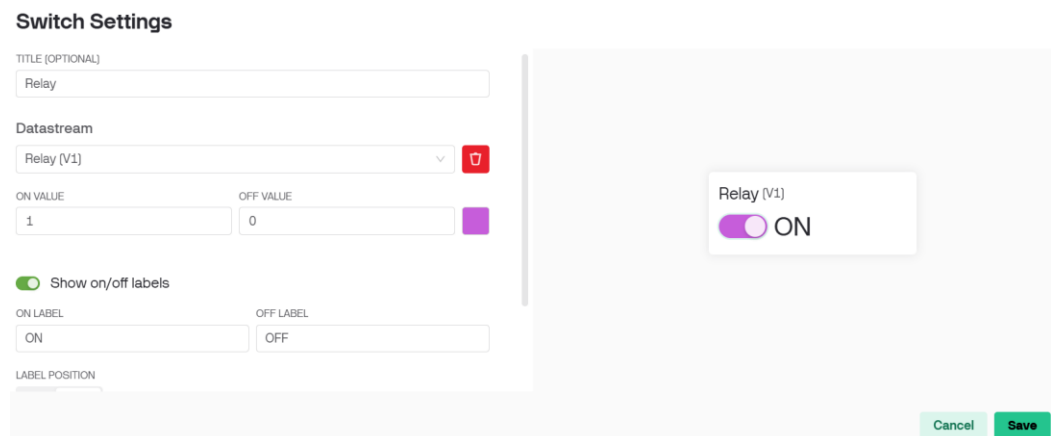
- g. Selanjutnya konfigurasi menu **Web Dashboard**, pilih **Widget** sesuai kebutuhan > lalu *drag and drop* ke **Dashboard** sebelah kanan.



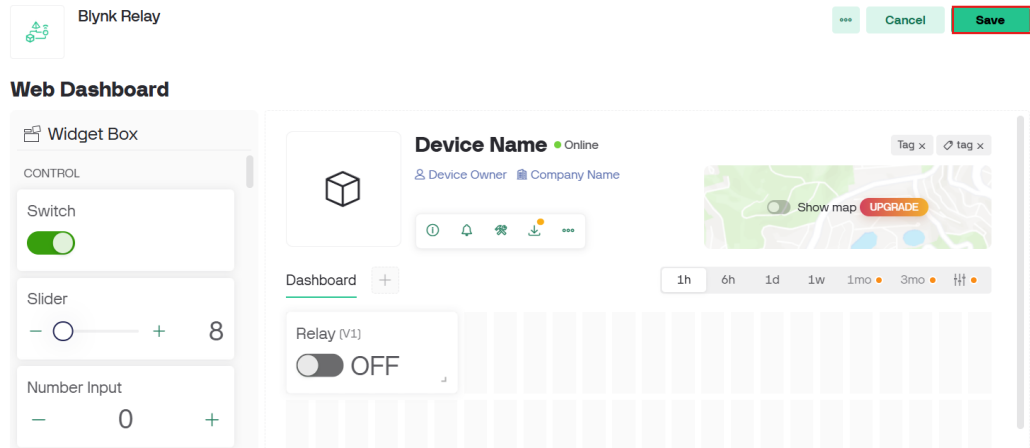
- h. Konfigurasi Widget **Switch** dengan klik ikon pengaturan seperti gambar.



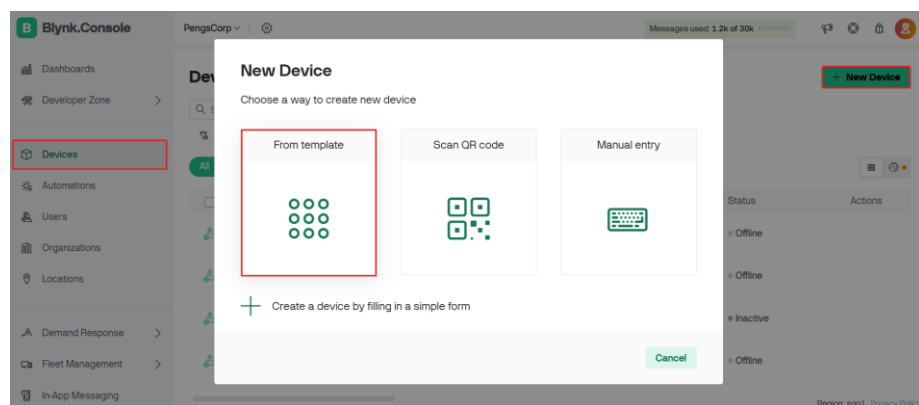
- i. Masukkan **TITLE** sesuai kebutuhan, ubah sumber **Datastream** menjadi **Relay (V1)** seperti yang telah dikonfigurasi. Kolom **ON VALUE** ubah menjadi 1 dan kolom **OFF VALUE** menjadi 0. Jika sudah klik **Save**.



- j. Konfigurasi **Web Dashboard** telah selesai, klik **Save**.



- k. Setelah konfigurasi Web Dashboard, selanjutnya menambahkan Device. Masuk ke menu **Devices** > klik **New Device** > pilih **From Template**.



1. Pada kolom **TEMPLATE** pilih template yang telah dikonfigurasi tadi (Blynk Relay).

New Device

Create new device by filling in the form below

TEMPLATE

DEVICE NAME
 11 / 50

Cancel Create

- m. Simpan kode berikut untuk dikonfigurasi di Arduino IDE.

```
New Device Created! ×
#define BLYNK_TEMPLATE_ID "TMPL6z9CHX-h7"
#define BLYNK_TEMPLATE_NAME "Blynk Relay"
#define BLYNK_AUTH_TOKEN "86_7DH38A9CnLvLY1cWavD8tU400iY-j"
```

- n. Selanjutnya konfigurasi program di Arduino IDE, masukkan kode berikut.

```
#define BLYNK_TEMPLATE_ID "TMPL6j44liV1G" // ID template
// dari Blynk
#define BLYNK_TEMPLATE_NAME "blynkrely" // Nama template
// Blynk
#define BLYNK_AUTH_TOKEN "9xtjeZJ-82CiVSKjWNDAXirniylY8ebv"
// Token autentikasi Blynk

#include <WiFi.h> // Library untuk koneksi WiFi (ESP32)
#include <BlynkSimpleEsp32.h> // Library Blynk untuk ESP32

// Konfigurasi WiFi
char ssid[] = "Nangka"; // SSID WiFi yang digunakan
char pass[] = "dikap999"; // Password WiFi

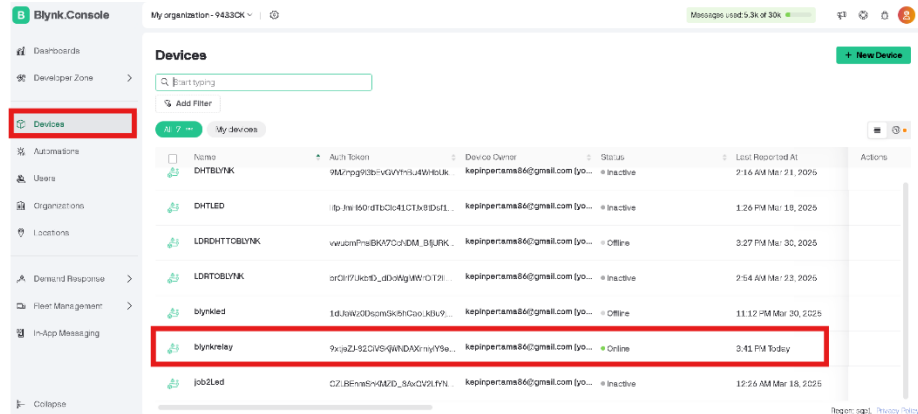
// Pin Relay
#define RELAY_PIN 12 // Pin GPIO 12 digunakan untuk
// mengontrol relay

void setup() {
  Serial.begin(115200); // Inisialisasi komunikasi serial
  Blynk.begin(BLYNK_AUTH_TOKEN, ssid, pass); //
  Menghubungkan ESP32 ke Blynk dan WiFi
  pinMode(RELAY_PIN, OUTPUT); // Atur pin relay sebagai
  output
  digitalWrite(RELAY_PIN, LOW); // Set relay dalam keadaan
  mati saat awal menyala
}

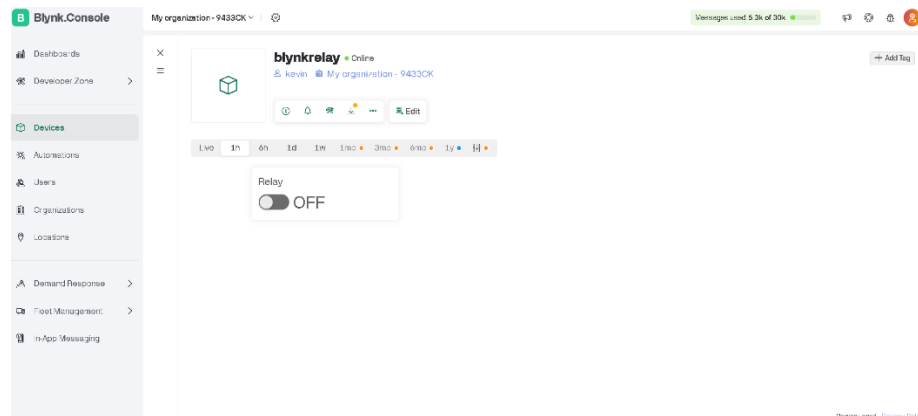
void loop() {
  Blynk.run(); // Proses utama untuk koneksi dan komunikasi
  dengan Blynk
}

// Fungsi kontrol relay dari aplikasi Blynk
BLYNK_WRITE(V1) {
  int status = param.asInt(); // Membaca nilai dari tombol
  di aplikasi (0 atau 1)
  digitalWrite(RELAY_PIN, status); // Atur status relay
  sesuai input dari Blynk
  Serial.print("Relay: ");
  Serial.println(status ? "NYALA" : "MATI"); // Tampilkan
  status relay di Serial Monitor
}
```

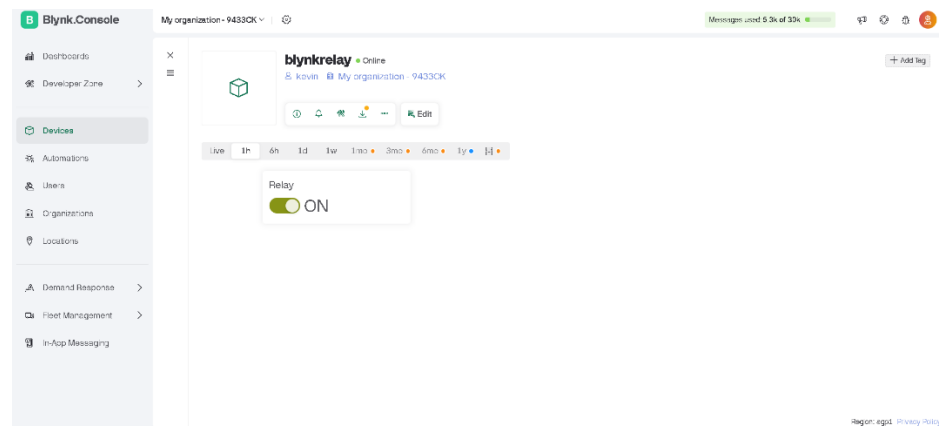
- o. Hubungkan ESP32 ke laptop, sebelum Verify atau Compile code pastikan Board ESP32 sudah terhubung di Arduino IDE, selanjutnya Upload.
- p. Periksa hasil melalui Blynk, pergi ke menu **Devices** lalu pilih **Relay Blynk**.



- q. Apabila sudah terhubung, maka status akan menjadi Online.

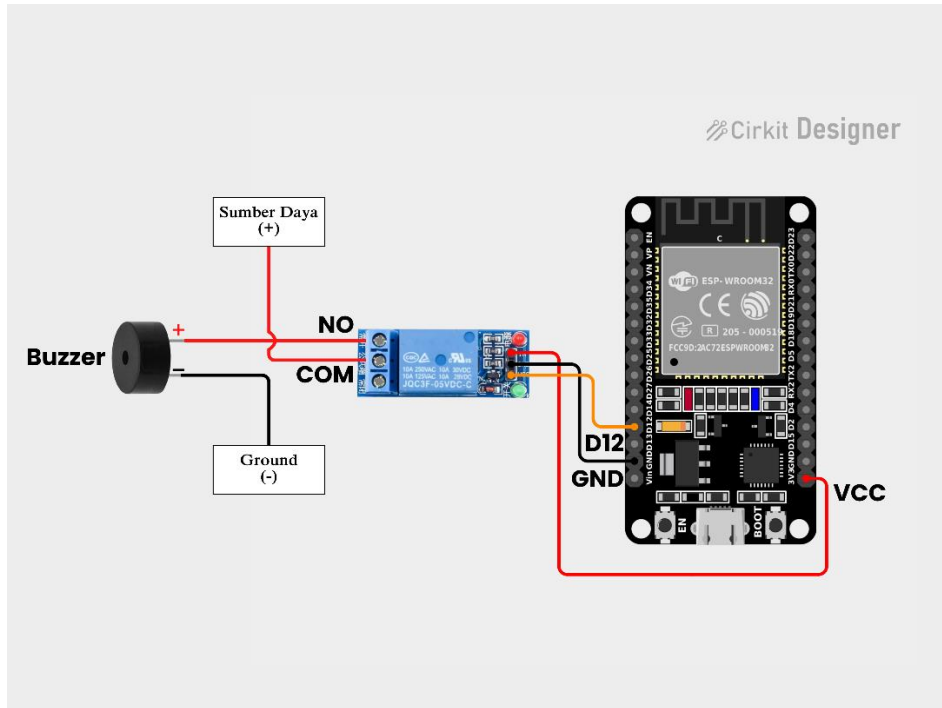


- r. Ketika Tombol switch diubah ke ON maka relay akan aktif dan sebaliknya.



F. Soal

1. Tampilkan output berdasarkan dari instruksi Langkah Kerja! Tambahkan analisis dan prinsip kerjanya di laporan praktikum!
2. Modifikasi rangkaian relay dengan kombinasi aktuator. Sehingga bisa mengontrol aktuator melalui relay! Gunakan aktuator LED atau Buzzer!



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/1X2R15dGXmTuoX5a81Ebzub6cuEzUz1AB?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