

# LAPORAN PRAKTIKUM

## TEKNIK ANTARMUKA

### MODUL VII

#### *INTERNET OF THINGS PLATFORMS*

Nama : Syarif Hidayat

NIM : D400220086

Kelas : A

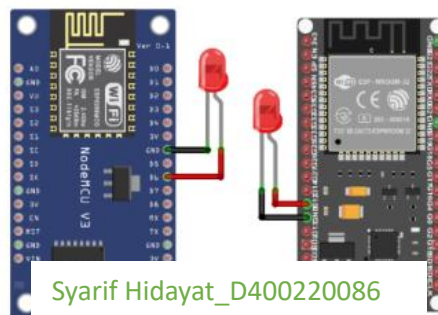
Tanggal Praktikum : 26 – November - 2024

Nilai :

#### A. Percobaan 1 : Kontrol LED ON/OFF Dengan Aplikasi Blynk

##### A.1 Hasil Percobaan

##### A.1.1 Gambar Rangkaian



##### A.1.2 Script

```
1 #define BLYNK_PRINT Serial
2 #define BLYNK_TEMPLATE_ID " TMPL6vQPsgzM3"
3 #define BLYNK_TEMPLATE_NAME "Control LED"
4 #if defined (ESP32)// Syarif Hidayat_D400220086
5 #include <WiFi.h>// Syarif Hidayat_D400220086
6 #include <BlynkSimpleEsp32.h>
7 #elif defined (ESP8266)
8 #include <ESP8266WiFi.h>
9 #include <BlynkSimpleEsp8266.h>
10 #endif// Syarif Hidayat_D400220086
11 char auth[] = " YMvrepfSpVb2f4tMOesEjzalhHZC1l0y";
12 char ssid[] = "TEBU IJO";
13 char pass[] = "selalubahagia";
14 const int ledPin = 12;// Syarif Hidayat_D400220086
15 void setup() { // Syarif Hidayat_D400220086
```

```

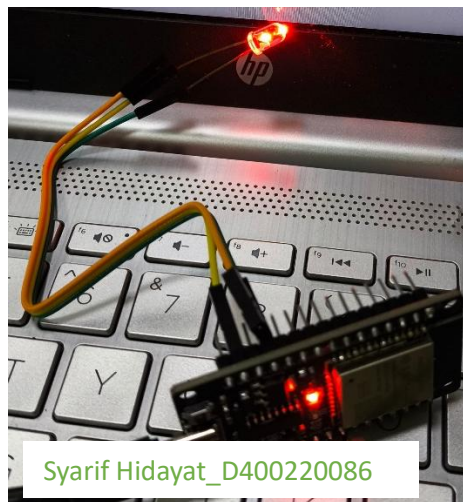
16 Serial.begin(115200);
17 Blynk.begin(auth, ssid, pass);
18 pinMode(ledPin, OUTPUT);
19 }// Syarif Hidayat_D400220086
20 void loop() {
21   Blynk.run();
22 }// Syarif Hidayat_D400220086
23 BLYNK_WRITE(V0) {
24   int buttonState = param.asInt();
25   if (buttonState == HIGH) {
26     digitalWrite(ledPin, HIGH);
27   } else { // Syarif Hidayat_D400220086
28     digitalWrite(ledPin, LOW);
29   } // Syarif Hidayat_D400220086
30 }

```

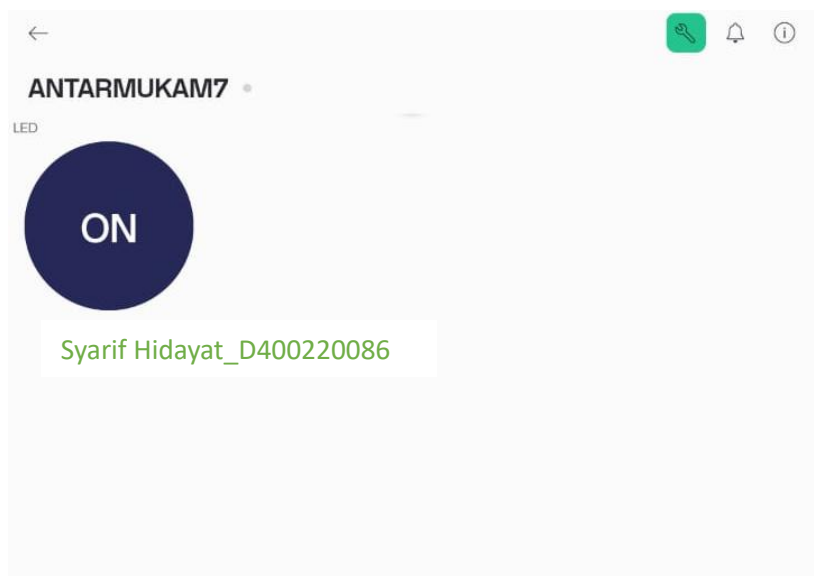
### A.1.3 Output

#### ➤ LED ON

- Foto Hardware

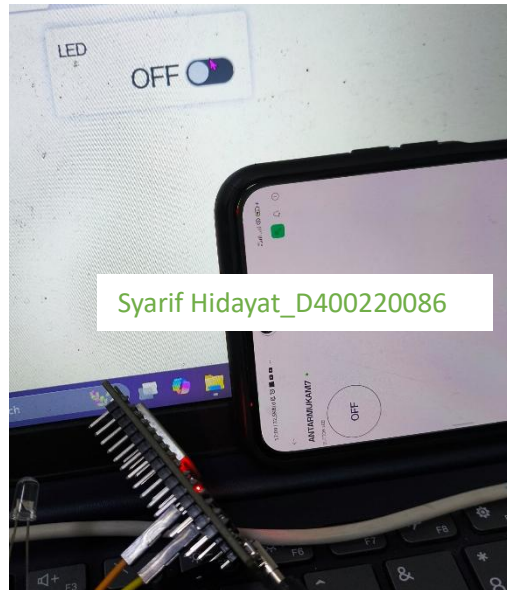


- Tampilan Blynk

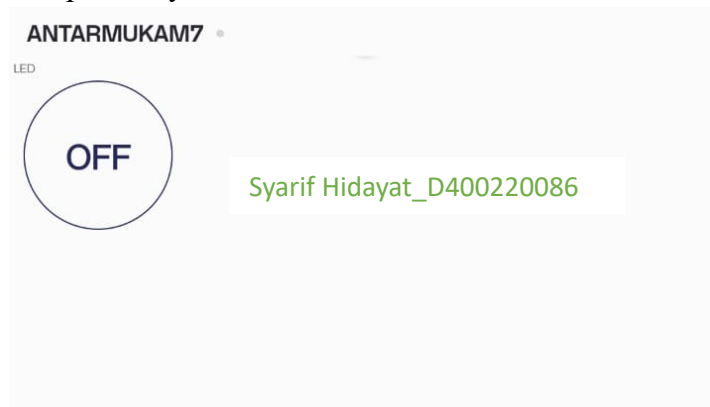


➤ LED OFF

- Foto Hardware



- Tampilan Blynk



## A.2 Pembahasan

### A.2.1 Pertanyaan

Tipe koneksi apa saja yang didukung oleh Platform Blynk?

Jelaskan maksud dari masing-masing tipe koneksi tersebut!

### A.2.2 Analisis

Blynk merupakan aplikasi yang bisa digunakan untuk mengontrol dan memonitoring sensor maupun aktuator. Pada percobaan ini LED yang tersambung oleh ESP32 dihubungkan dengan blynk agar bisa dikontrol secara online dan bisa juga dikontrol dari jarak jauh. Sebelum LED bisa dikontrol secara online dan jarak jauh maka ada beberapa langkah-langkah yang perlu dilakukan terlebih dahulu yang pertama menghubungkan ESP32 dengan WiFi ataupun hotspot dengan cara memasukkan ssid beserta password WiFi atau hotspot pada program dan juga memanggil library WiFi pada program, langkah yang selanjutnya itu membuat tampilan pada blynk

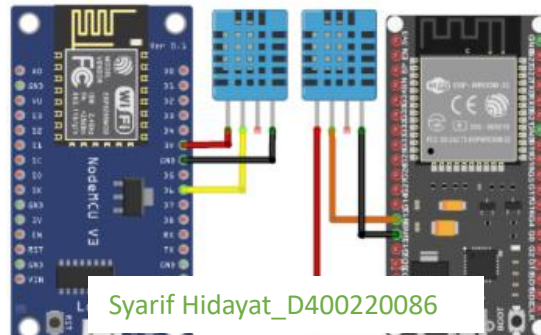
agar bisa mengontrol LED pada blynk menambahkan widget BUTTON kemudian mengatur BUTTON tersebut menjadi pin virtual kemudian mengubah mode BOTTON tersebut menjadi switch langkah selanjutnya yaitu mengcopy auth dari tampilan blynk tersebut dan di paste pada program. Jika auth tidak sama dengan yang ada ditampilan yang sudah dibuat maka LED tidak akan bisa dikontrol. Didalam fungsi setup harus memanggil ssid, password dan auth, tambahan jika menggunakan server selain server asli dari blynk maka pada fungsi setup harus menambahkan server apa yang digunakan dan port berapa yang digunakan(Blynk.begin(auth, ssid, password, "server", port)). Membuat fungsi BLYNK\_WRITE(VO) menandakan fungsi virtual ini yang akan dipanggil ketika program dijalankan fungsi ini yang nantinya akan mengetahui apakah tombol pada blynk dipencet atau tidak. Cara kerja fungsi ini yaitu dengan menerima nilai HIGH atau LOW dari param.asInt() jika param.asInt() menerima nilai HIGH maka LED yang terhubung dengan ESP32 akan menyala dan jika param.asInt() menerima nilai LOW maka LED akan mati dan fungsi yang paling penting adalah Blynk.run() jika pada program tidak terdapat fungsi ini maka ESP32 yang sudah terhubung dengan WiFi tidak akan bisa berjalan meskipun sudah memasukkan auth pada program.

Blynk mendukung berbagai macam koneksi yang paling umum digunakan adalah WiFi yang terhubung oleh router agar bisa diakses oleh perangkat IoT dan bisa juga dengan hotspot. Ethernet tipe koneksi internet yang menggunakan kabel untuk mengirim jaringan, Ethernet juga memiliki jaringan yang lebih cepat dibandingkan Wi-Fi ataupun hotspot dan juga jika sinyal WiFi tidak sampai maka bisa menggunakan Ethernet. Lora Wan bisa juga digunakan untuk menghubungkan perangkat IoT dengan internet, digunakan untuk perangkat keras yang membutuhkan daya yang rendah dan jangkauan yang luas. Bluetooth juga salah satu koneksi yang bisa digunakan blynk tetapi koneksi ini memiliki jangkauan yang sangat kecil dan harus dekat dengan perangkat IOT dan memungkinkan pengontrolan atau memonitor secara langsung dengan pengguna.

## B. Percobaan 2 : Monitoring Sensor DHT11 Dengan Aplikasi Blynk

### B.1 Hasil Percobaan

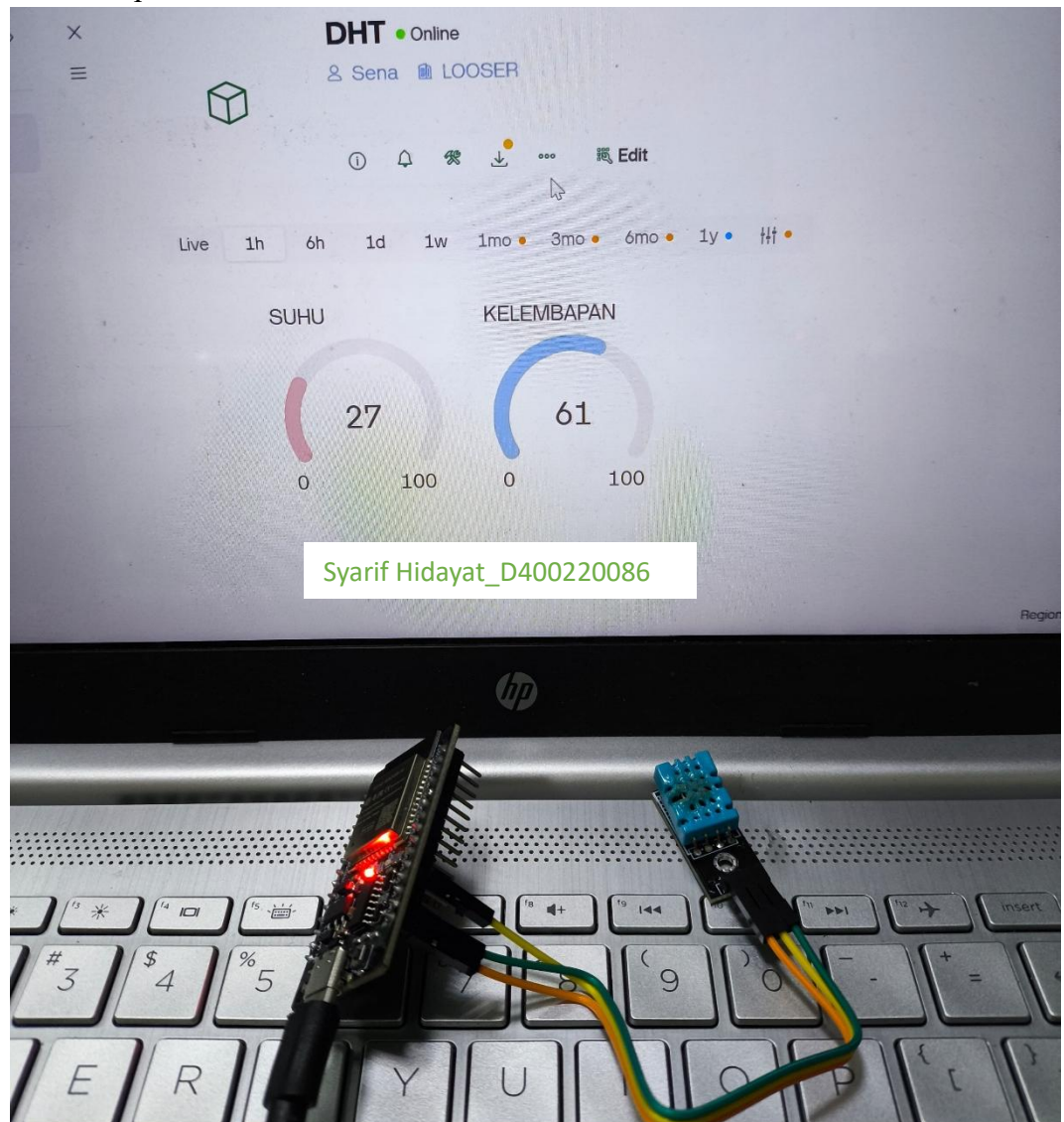
#### B.1.1 Gambar Rangkaian



#### B.1.2 Script

```
1 #define BLYNK_PRINT Serial
2 #define BLYNK_TEMPLATE_ID "TMPL6hoFq3-aX"
3 #define BLYNK_TEMPLATE_NAME "Monitoring DHT11"
4 #if defined(ESP32) //Syarif Hidayat_D400220086
5 #include <WiFi.h>
6 #include <BlynkSimpleEsp32.h>
7 #elif defined(ESP8266)
8 #include <ESP8266WiFi.h>
9 #include <BlynkSimpleEsp8266.h>
10 #endif //Syarif Hidayat_D400220086
11 #include <DHT.h>
12 char auth[] = "-TwRaJp8NSiWR-DcBM5WEgAika9zcPwp";
13 char ssid[] = "TEBU IJO";
14 char pass[] = "selalubahagia";
15 const int dhtPin = 12;
16 DHT dht(dhtPin, DHT11);
17 void setup() { //Syarif Hidayat_D400220086
18   Serial.begin(115200);
19   Blynk.begin(auth, ssid, pass);
20   dht.begin();
21 } //Syarif Hidayat_D400220086
22 void loop() {
23   Blynk.run();
24   // Baca nilai suhu dan kelembapan dari sensor DHT11
25   float suhu = dht.readTemperature();
26   float kelembapan = dht.readHumidity();
27   // Kirim nilai suhu dan kelembapan ke aplikasi Blynk
28   Blynk.virtualWrite(V1, suhu);
29   Blynk.virtualWrite(V2, kelembapan);
30 } //Syarif Hidayat_D400220086
```

### B.1.3 Output



## B.2 Pembahasan

### B.2.1 Pertanyaan

Jelaskan secara lengkap dari segi perubahan, bagaimana pembacaan nilai suhu dan kelembapan pada sensor DHT11 dapat bekerja?

### B.2.2 Analisis

Sama dengan percobaan pertama menggunakan blynk tetapi pada percobaan ini blynk digunakan untuk memonitoring suhu dan kelembapan menggunakan sensor DHT11. Sensor DHT11 yang dihubungkan dengan ESP32 dan terhubung dengan WiFi bisa digunakan untuk memonitoring pada blynk secara real-time. Sebelum bisa memonitoring suhu dan kelembapan maka ada beberapa langkah-langkah yang perlu dilakukan terlebih dahulu yang pertama menghubungkan ESP32 dengan WiFi ataupun hotspot dengan cara memasukkan ssid beserta password WiFi atau hotspot pada program

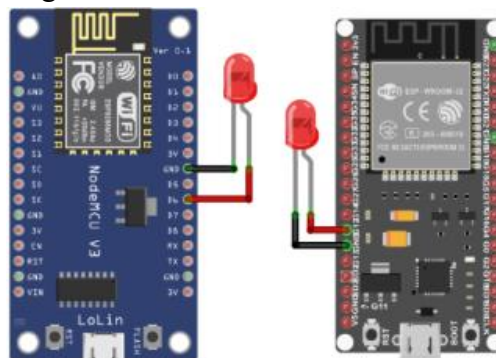
dan juga memanggil library WiFi pada program, langkah yang selanjutnya itu membuat tampilan pada blynk agar bisa memonitoring sensor DHT11 pada blynk menambahkan widget gauge karna pada percobaan ini memonitoring 2 hal maka kita memberikan 2 widget gauge untuk memonitoringnya. Setiap gauge harus memiliki 2 pin virtual yang berbeda-beda untuk memonitoring suhu menggunakan pin virtual V1 dan untuk memonitoring kelembapan menggunakan pin virtual V2, setiap gauge disetting untuk data analognya yaitu dari 0-255 selanjutnya yaitu mengcopy auth dari tampilan blynk tersebut dan di paste pada program, setiap kita membuat tampilan baru pada blynk maka auth nya akan berubah lagi. Pada program kita memanggil library dari sensor DHT11 dan mendefinisikannya. Didalam fungsi setup harus memanggil ssid, password dan auth, tambahan jika menggunakan server selain server asli dari blynk maka pada fungsi setup harus menambahkan server apa yang digunakan dan port berapa yang digunakan(Blynk.begin(auth, ssid, password, "server", port)). Dan juga memanggil sensor DHT11 yang sudah didefinisikan sebelumnya. Pada fungsi loop membuat 2 variabel, variabel pertama untuk menampung data suhu dari sensor DHT11 dan variabel yang kedua untuk menampung data kelembapan dari sensor DHT11. Blynk.virtual Write digunakan untuk mengirim data yang diterima dari kedua variabel ke blynk, kedua data tersebut dikirim ke virtual pin yang berbeda-beda. Blynk.run() digunakan untuk memastikan apakah ESP32 terhubung atau tidak.

DHT11 membaca nilai suhu dan kelembapan dengan menggunakan perubahan dari kapasitansi dan resistansi, Nilai-nilai ini nantinya akan dikonversi ke sinyal digital. Pembacaan suhu resistansinya akan menurun seiring kenaikan suhunya dan pembacaan nilai kelembapan kapasitansinya akan naik terus seiring dengan meningkatnya nilai kelembapan.

### C. Percobaan 3 : Kontrol LED ON/OFF Dengan Thingier.io

#### C.1 Hasil Percobaan

##### C.1.1 Gambar Rangkaian





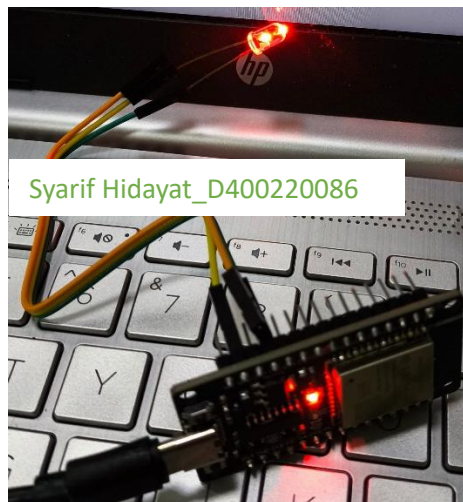
### C.1.2 Script

```
1 #if defined(ESP8266)
2 #include <ThingrESP8266.h>
3 #elif defined(ESP32)
4 #include <ThingrESP32.h>
5 #endif
6 //Syarif Hidayat_D400220086
7 #define USERNAME "TAMM"
8 #define DEVICE_ID "Control_LED"
9 #define DEVICE_CREDENTIAL "L@d-j5wzUbAajGUF"
10 #define SSID "RS">//Syarif Hidayat_D400220086
11 #define SSID_PASSWORD "RIFSYA16022"
12 #ifdef ESP8266ThingrESP8266 thing(USERNAME, DEVICE_ID, DEVICE_CREDENTIAL);
13 #elif ESP32ThingrESP32 thing(USERNAME, DEVICE_ID, DEVICE_CREDENTIAL);
14 #endif
15 //Syarif Hidayat_D400220086
16 int ledPin = 12;
17 void setup() {
18   pinMode(ledPin, OUTPUT);
19   thing.add_wifi(SSID, SSID_PASSWORD);
20   // Registrasi fungsi untuk mengontrol LED
21   thing["LED"] << digitalPin(ledPin);
22 }//Syarif Hidayat_D400220086
23 void loop() {
24   // Handle koneksi dan komunikasi Thingr.io
25   thing.handle();
26 }//Syarif Hidayat_D400220086
27
```

### C.1.3 Output

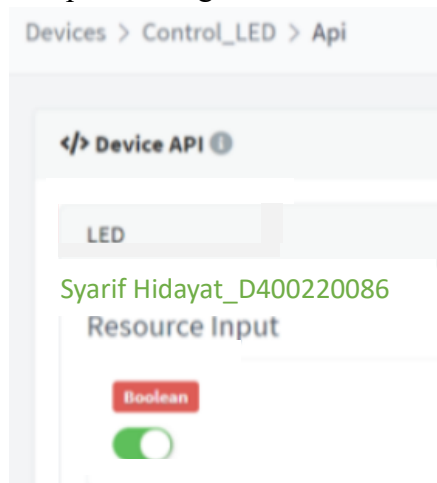
#### ➤ LED ON

- Foto Hardware



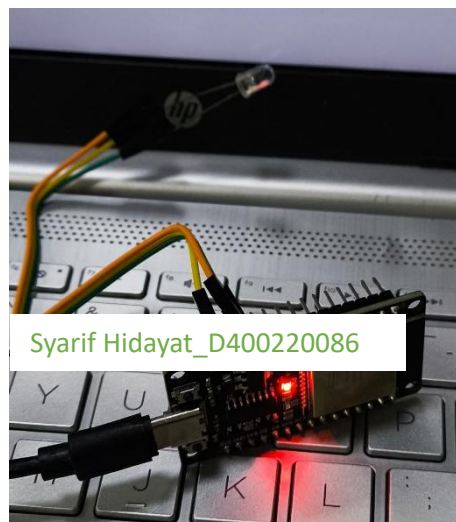


- Tampilan Thinger.io

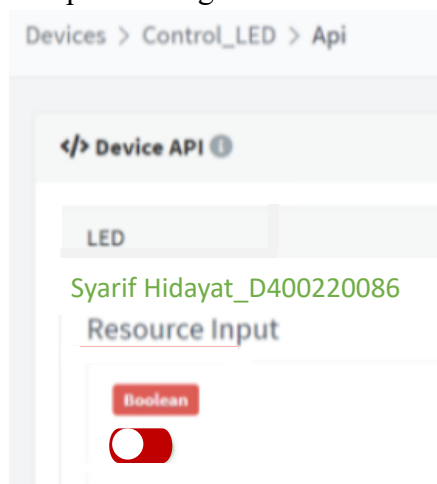


➤ LED OFF

- Foto Hardware



- Tampilan Thinger.io



## C.2 Pembahasan

### C.2.1 Pertanyaan

Tipe device apa saja yang didukung oleh platform Thinger.io pada saat berada pada laman Device Configuration? Jelaskan maksud dari masing-masing tipe tersebut!

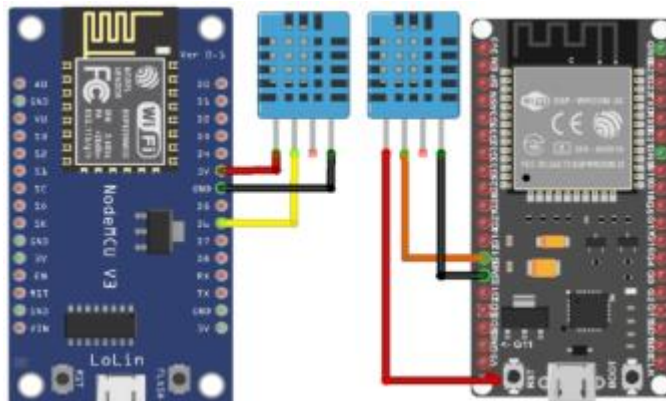
### C.2.2 Analisis

Pada percobaan ini LED yang tersambung oleh ESP32 dihubungkan dengan Thinger.io agar bisa dikontrol secara online dan bisa juga dikontrol dari jarak jauh. Sebelum LED bisa dikontrol secara online dan jarak jauh maka ada beberapa langkah-langkah yang perlu dilakukan terlebih dahulu yang pertama menghubungkan ESP32 dengan WiFi ataupun hotspot dengan cara memasukkan ssid beserta password WiFi atau hotspot pada program dan juga memanggil library thinger.io pada program dan juga memanggil fungsi untuk menghubungkan esp32 dengan Thinger.io. Mendefinisikan pin LED, Didalam fungsi setup harus memanggil ssid, password dari WiFi kemudian mendeklarasikan pin LED yang sudah didefinisikan dan juga digitalPin digunakan untuk membaca dan menulis pin digital, fungsi yang paling penting adalah thing.handle() berfungsi untuk menangani pull dan request dari Thinger.io untuk mengontrol LED. Platform Thinger.io mendukung beberapa device yang pertama adalah Generic device asalkan terhubung dengan koneksi internet dan dapat pull dan request data maka device ini bisa digunakan, MQTT device dimana device ini digunakan untuk perangkat yang memiliki daya terbatas dan juga menggunakan protokol MQTT, HTTP device dimana tipe ini menggunakan protokol komunikasi HTTP untuk berkomunikasi dengan platform Thinger.io

## D. Percobaan 4 : Monitoring Sensor DHT11 Dengan Thinger.io

### D.1 Hasil Percobaan

#### D.1.1 Gambar Rangkaian

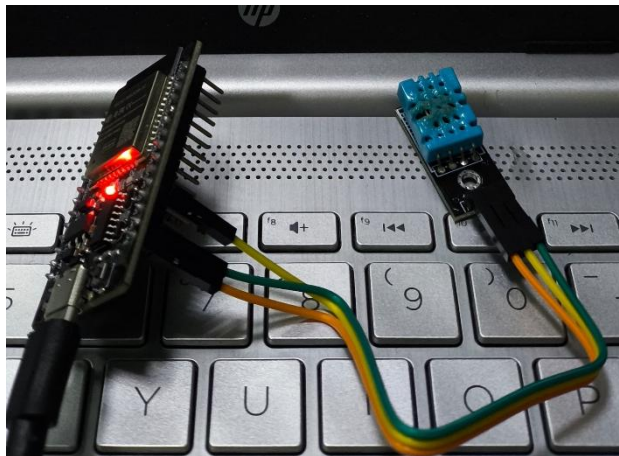


### D.1.2 Script

```
1 #if defined(ESP8266)
2 #include <ThingierESP8266.h>
3 #elif defined(ESP32)
4 #include <ThingierESP32.h>
5 #endif //Syarif Hidayat_D400220086
6 #include <DHT.h>
7 #define USERNAME "Vikaaleyda"
8 #define DEVICE_ID "Monitoring_DHT11"
9 #define DEVICE_CREDENTIAL "QA_DhCvHUdBSrrmn"
10 #define SSID "TEBU IJO"
11 #define SSID_PASSWORD "selalubahagia"
12 #ifdef ESP8266 //Syarif Hidayat_D400220086
13 ThingierESP8266 thing(USERNAME, DEVICE_ID, DEVICE_CREDENTIAL);
14 #elif ESP32 //Syarif Hidayat_D400220086
15 ThingierESP32 thing(USERNAME, DEVICE_ID, DEVICE_CREDENTIAL);
16 #endif //Syarif Hidayat_D400220086
17 #define DHT_PIN 12
18 DHT dht(DHT_PIN, DHT11);
19 float Humidity, Temperature;
20 void setup() { //Syarif Hidayat_D400220086
21   thing.add_wifi(SSID, SSID_PASSWORD);
22   dht.begin(); //Syarif Hidayat_D400220086
23   thing["DHT11"] >> [] (pson & out) {
24     out["Humidity"] = dht.readHumidity();
25     out["Celsius"] = dht.readTemperature();
26   }; //Syarif Hidayat_D400220086
27 } //Syarif Hidayat_D400220086
28 void loop() {
29   thing.handle();
30   Humidity = dht.readHumidity();
31   Temperature = dht.readTemperature();
32 } //Syarif Hidayat_D400220086
```

### D.1.3 Output

- Hardware



- Tampilan Thingier.io



## D.2 Pembahasan

### D.2.1 Pertanyaan

Apa fungsi dari Device Credentials pada platform Thinger.io? Apa yang jika tidak menggunakan credentials? Jelaskan!

### D.2.2 Analisis

Percobaan ini Thinger.io digunakan untuk memonitoring suhu dan kelembapan menggunakan sensor DHT11. Sensor DHT11 yang dihubungkan dengan ESP32 dan terhubung dengan WiFi bisa digunakan untuk memonitoring pada Thinger.io secara real-time. Sebelum bisa memonitoring suhu dan kelembapan maka ada beberapa langkah-langkah yang perlu dilakukan terlebih dahulu yang pertama menghubungkan ESP32 dengan WiFi ataupun hotspot dengan cara memasukkan ssid beserta password WiFi atau hotspot pada program dan juga memanggil library thinger.io pada program, langkah yang selanjutnya itu membuat tampilan pada thinger.io agar bisa memonitoring sensor DHT11 pada Thinger.io menambahkan 2 donut chart untuk memonitoringnya. Setiap donut chart harus memiliki chart yang berbeda-beda untuk memonitoring suhu dan untuk memonitoring kelembapan. Pada program kita memanggil library dari sensor DHT11 dan mendefinisikannya. Didalam fungsi setup memanggil ssid, password, juga memanggil sensor DHT11 yang sudah didefinisikan sebelumnya dan fungsi out digunakan untuk mengirim data ke Thinger.io. Pada fungsi loop membuat 2 variabel, variabel pertama untuk menampung data suhu dari sensor DHT11 dan variabel yang kedua untuk menampung data kelembapan dari sensor DHT11.

Device Credentials pada platform Thinger.io bertujuan untuk mengotentikasi perangkat ke Thinger.io, jika tidak menggunakan device Credentials maka perangkat IoT tidak akan bisa terhubung dengan Thinger.io dan juga tidak akan bisa mengirim dan menerima data dari Thinger.io.

#### E. Foto Kelompok



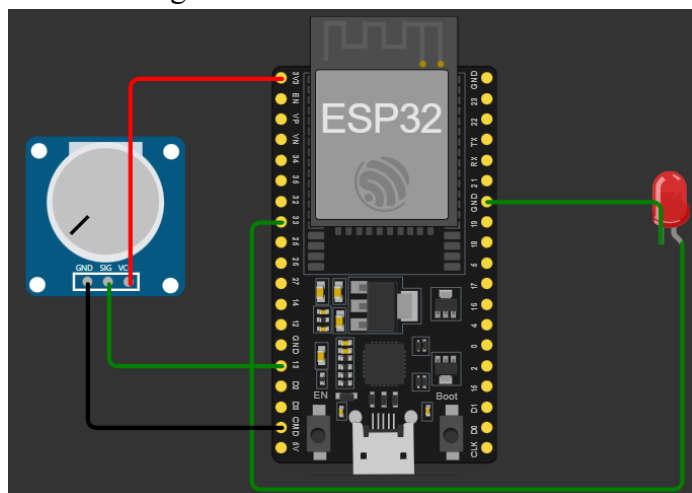
#### F. Pertanyaan

Buatlah sebuah proyek untuk mengontrol kecerahan LED menggunakan Potensiometer dengan platforms *Internet Of Things*. Instruksi lengkap sebagai berikut :

- Gunakan 1 LED dan 1 potensiometer.
- Platform IoT dapat menggunakan Blynk atau Thingier.io.
- Potensiometer mempengaruhi kecerahan LED.
- Nilai Potensiometer tertampil pada platform IoT.
- Pada platform IoT terdapat keterangan 3 kondisi LED, yaitu Gelap, Redup dan Terang.
- Dalam pengerjaan menggunakan hardware secara langsung.

Jawab :

- Gambar Rangkaian



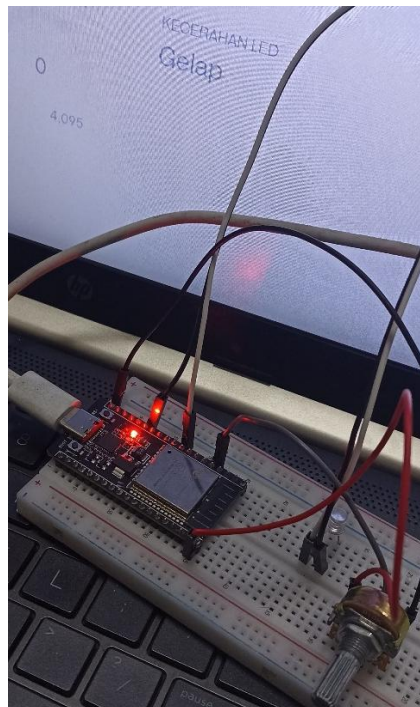
- Script

```
1 #include <WiFi.h> //Syarif Hidayat_D400220086
2 #include <BlynkSimpleEsp32.h>
3 char ssid[] = "RS";
4 char pass[] = "Rifsya160022";
5 char auth[] = "d4c26f00b7c5400ab508b9414b74d59f";
6 void setup() { //Syarif Hidayat_D400220086
7     Serial.begin(115200);
8     pinMode(12, INPUT);
9     pinMode(33, OUTPUT);
10    Blynk.begin(auth, ssid, pass, "blynk.ipnex.co.th", 8080);
11 } //Syarif Hidayat_D400220086
12 void loop() {
13     Blynk.run(); //Syarif Hidayat_D400220086
14     int pot = analogRead(33);
15     pot = map(pot, 0, 4095, 0, 1023);
16     Serial.print("Pot : ");
17     Serial.println(pot);
18     Blynk.virtualWrite(V0, pot);
19     int com = map(pot, 0, 1023, 0, 355);
20     if (com < 55) { //Syarif Hidayat_D400220086
21         Blynk.virtualWrite(V4, "Gelap");
22     } else if (com >= 55 && com < 170) {
23         Blynk.virtualWrite(V4, "Mendung");
24     } else { //Syarif Hidayat_D400220086
25         Blynk.virtualWrite(V4, "Terang");
26     }
27 } //Syarif Hidayat_D400220086
```

- Hasil

- Gelap

- Foto Hardware

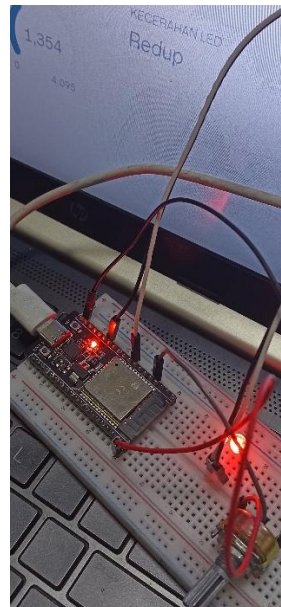


➤ Tampilan Platform



○ Redup

➤ Foto Hardware

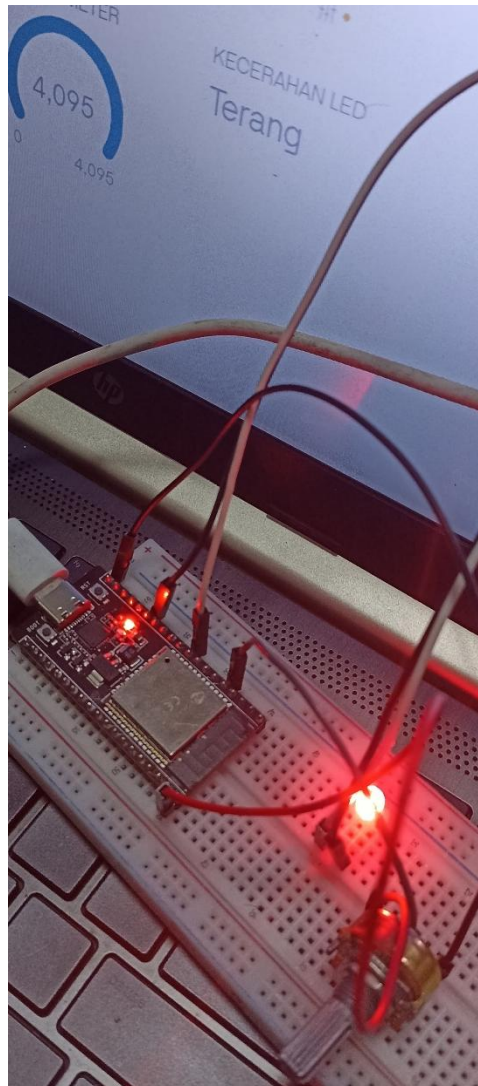


➤ Tampilan Platform





- Terang
  - Foto Hardware



- Tampilan Platform



#### G. Kesimpulan

- Blynk dan Thinger.io merupakan platform IoT yang memiliki banyak fungsi bisa digunakan untuk mengontrol dan juga untuk memonitoring sensor maupun aktuator.
- Setiap membuat tampilan baru pada blynk maka kita akan mendapatkan auth yang berbeda dari yang sebelumnya.
- Ketika membuat new device pada Thinger.io maka kita akan diberikan random Credentials setiap kali kita ingin melihat random Credentials maka kita akan diberikan Credentials yang baru lagi.
- Jika kita menggunakan blynk maka kita harus menambahkan blynk.run pada program jika menggunakan thinger.io maka kita harus menambahkan thing.handle.
- Setiap platform IoT memiliki cara menghubungkan yang berbeda beda jika menggunakan blynk maka pada setup kita harus memanggil ssid, password dan auth secara barengan dan jika kita menggunakan Thinger.io maka kita memanggil Credentials sebelum fungsi setup.

Telah diperiksa oleh,

Dzaky Novika Ramadhan

NIM. D400210094

(.....)

Dedi Ary Prasetya S.T., M.Eng

NIDN. 615117504

(.....)