

ESP32 IoT

Mode Fast Track

Modul Ini Merupakan Modul Sample Digunakan
Untuk Pelatihan IoT Sederhana

(Oleh: Wahyu Firmansyah)



ESPRESSIF



DAFTAR ISI

Basic ESP32.....	3
Pengenalan ESP32	3
Produk ESPRESSIF	3
Pin Out ESP32	3
Instalasi Varian ESP Arduino IDE	4
Telegram BOT	6
Pengenalan	6
Fungsi Telegram BOT	6
Pembuatan BOT.....	6
Protokol.....	7
Library.....	7
Program.....	8
Hasil Program	9
Blynk Platfrom.....	10
Pengenalan	10
Fungsi Blynk.....	10
Instalasi Blynk	10
Konfigurasi Blynk.....	11
Protokol.....	12
Library.....	12
Program.....	13
Hasil Program	14
HTTP.....	16
Pengenalan	16
Debugging.....	16
Program.....	17
Hasil Program	18

Basic ESP32

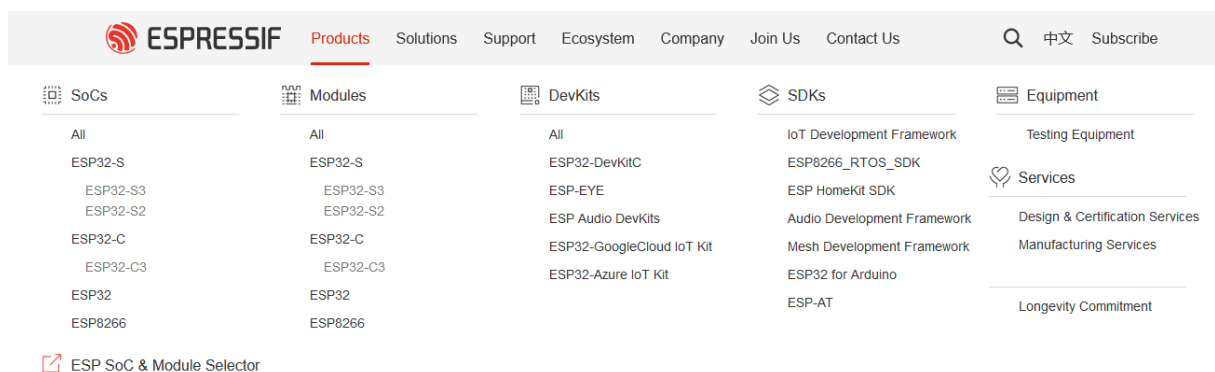
Pengenalan ESP32



ESP32 merupakan salah satu varian produk dari **Espressif System**. Produk ini sendiri adalah salah satu contoh tipe mikrokontroler yang dapat digunakan untuk belajar tentang perangkat IoT.

Produk ESPRESSIF

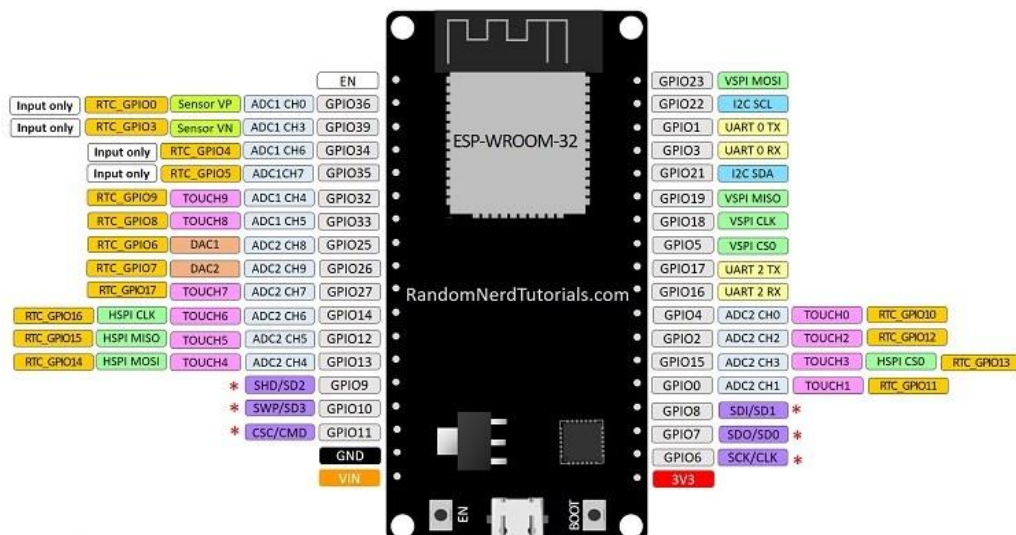
Berikut adalah macam macam product dan services dari **Espressif System**.



Pin Out ESP32

Menampilkan pin apa saja yang tersedia dalam ESP32.

ESP32 DEVKIT V1 – DOIT version with 36 GPIOs

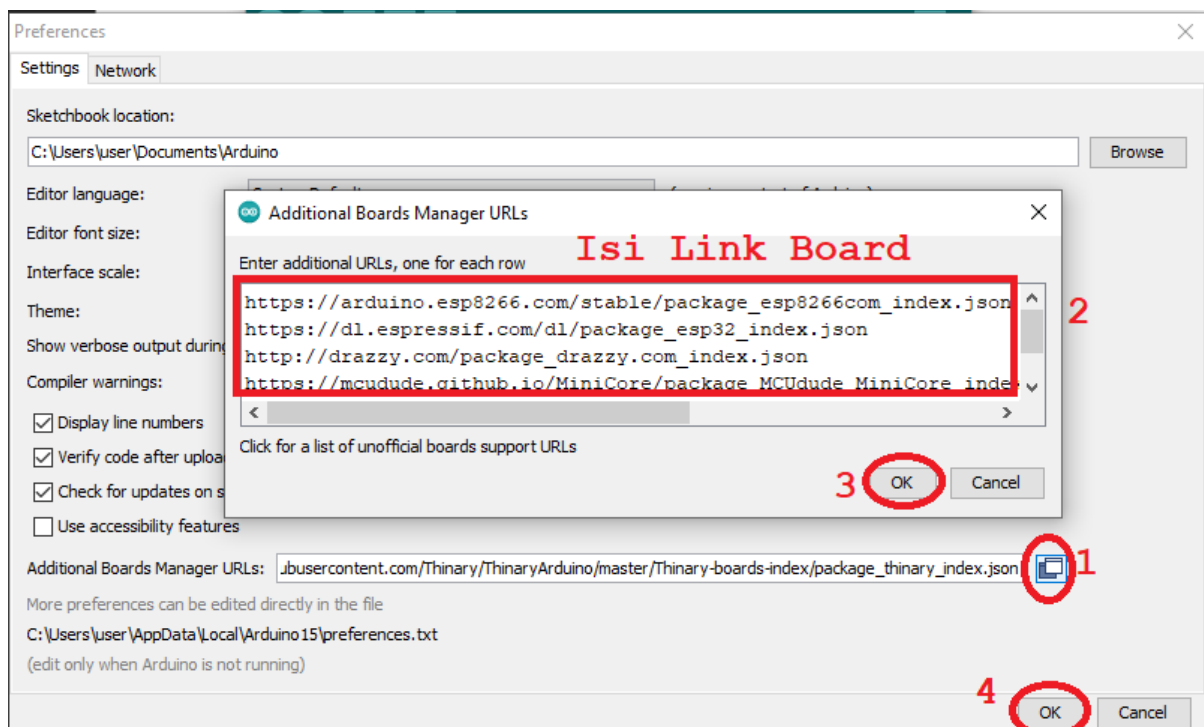
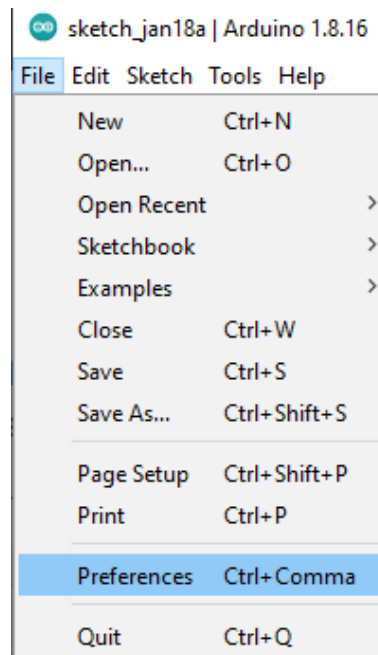


* Pins SCK/CLK, SDO/SD0, SDI/SD1, SHD/SD2, SWP/SD3 and SC5/CMD, namely, GPIO6 to GPIO11 are connected to the integrated SPI flash integrated on ESP-WROOM-32 and are not recommended for other uses.

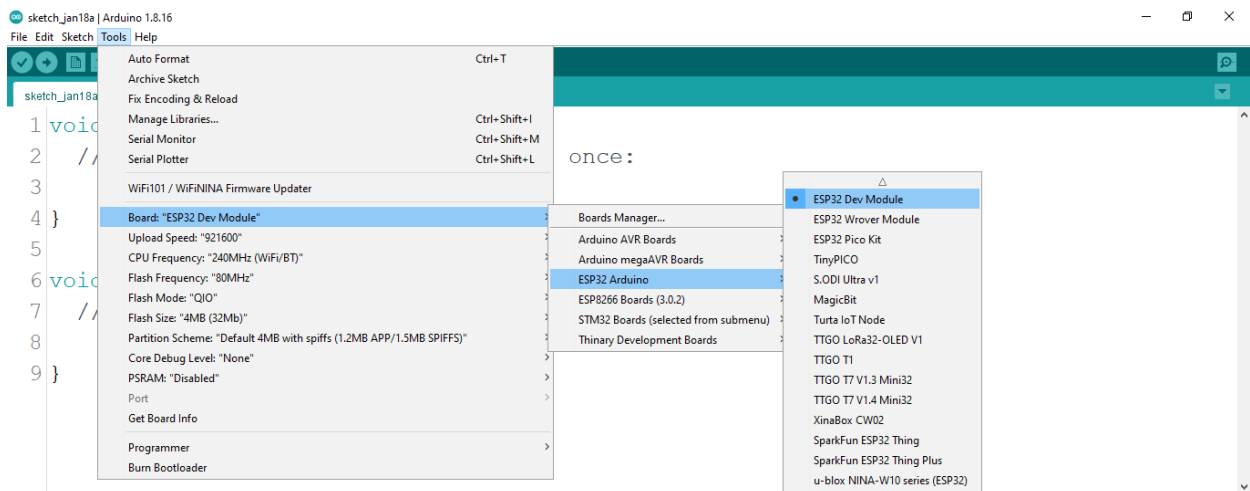
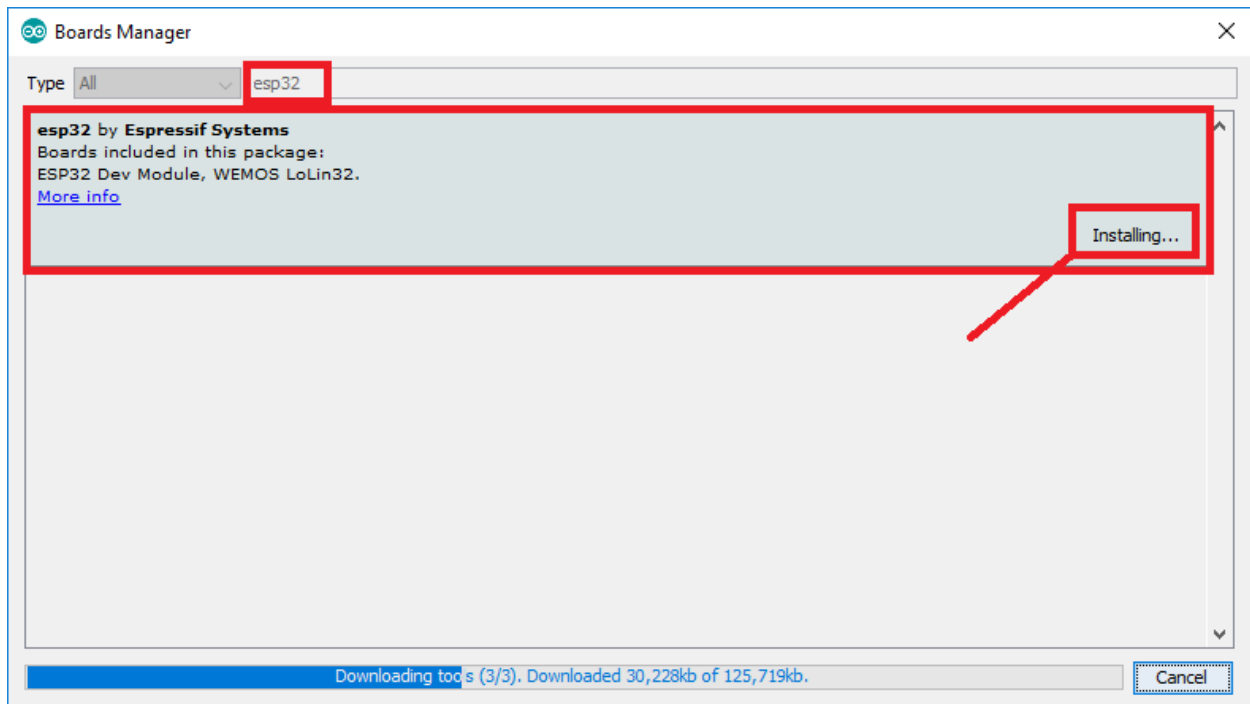
Source : <https://github.com/bywahjoe/kelasIOT>

Instalasi Varian ESP Arduino IDE

Cara instalasi varian ESP di Arduino IDE



```
https://dl.espressif.com/dl/package_esp32_index.json  
https://arduino.esp8266.com/stable/package_esp8266com_index.json
```



Setelah berhasil instalasi pada board ESP32 pada Arduino IDE klik **Tools > Board > ESP**.Pengecekan ini dilakukan apakah **Board ESP** sudah masuk dan terinstal tanpa error.

Telegram BOT

Pengenalan





Telegram merupakan salah satu contoh platform messaging yang dapat berjalan pada aplikasi smartphone maupun desktop. Aplikasi ini juga menyediakan Telegram BOT dan API BOT secara gratis dan mudah.

Fungsi Telegram BOT

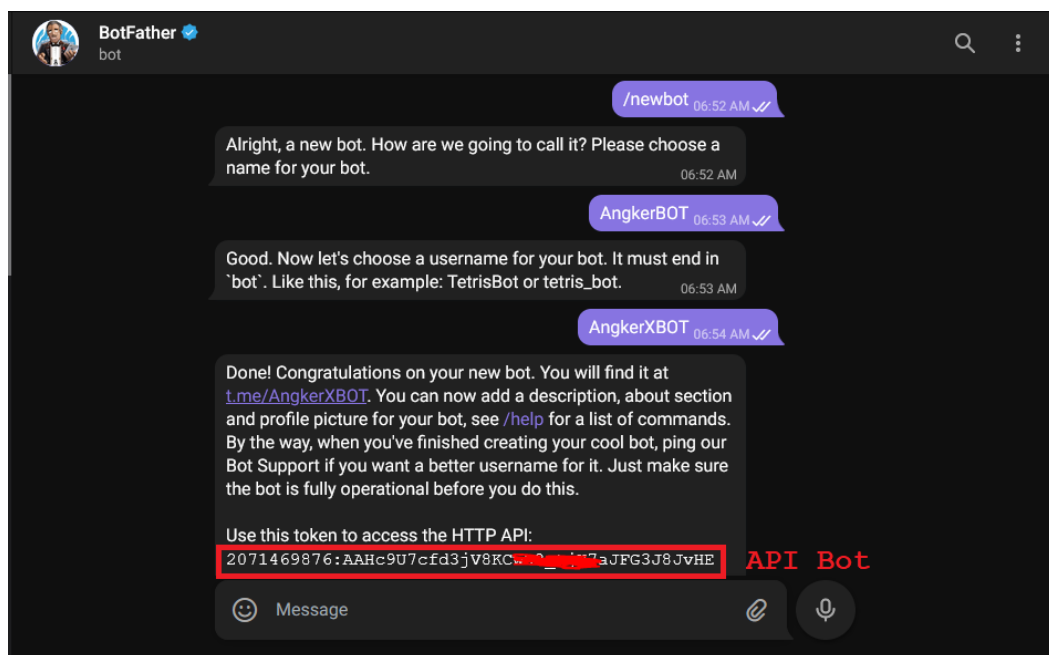
Fungsi dari telegram bot sendiri biasanya untuk monitoring pada server, notifikasi urgent, callback, message broadcast dan lain lain.

Pembuatan BOT

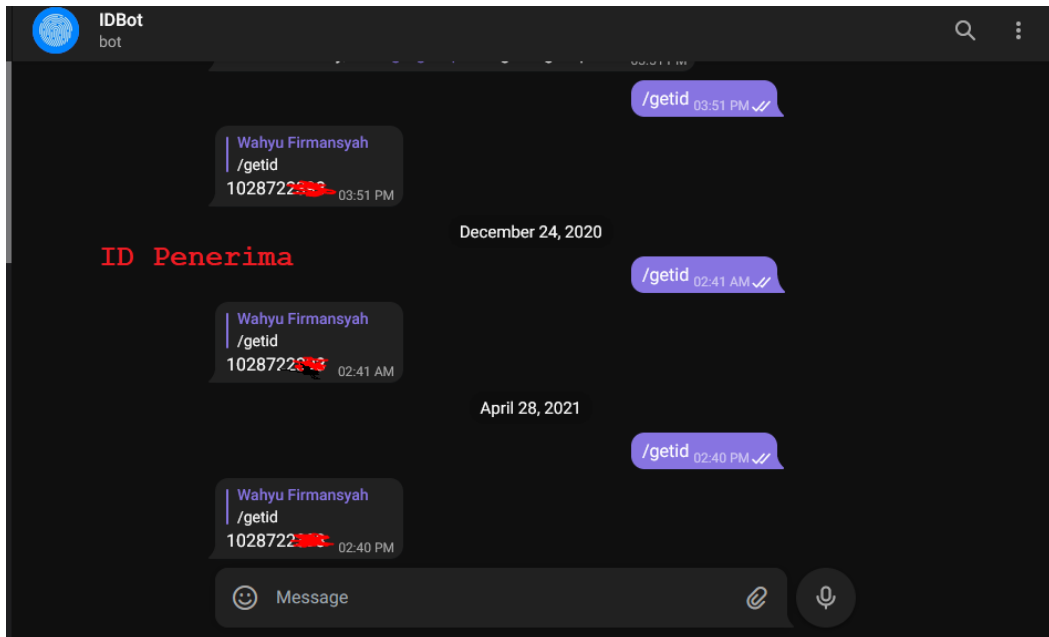
Langkah selanjutnya kita akan membuat custom bot.

 BotFather @BotFather	BotFather sendiri adalah pengelola akun bot yang akan kita buat. Akun ini digunakan untuk mengolah data seperti nama bot, username, API BOT .
 IDBot @myidbot	Melalui IDBot kita dapat mengetahui ID Akun yang akan kita jadikan sebagai tujuan penerima.

- Berikut adalah tutorial menggunakan BotFather untuk membuat BOT, setting nama BOT, mendapatkan API.



- Berikut adalah tutorial menggunakan **IDBot** untuk mendapatkan ID tujuan yang akan menerima pesan.



Setelah melalui tahap diatas kita otomatis sudah mendapatkan konfigurasi BOT yang sudah kita buat seperti:

1. Username BOT
2. Nama BOT
3. **Token API BOT** [Catat Token]
4. **ID Tujuan** [Catat ID Tujuan]

Protokol

Untuk komunikasinya sendiri **ESP32** mengambil dan menerima data menggunakan protokol **HTTP** dengan kombinasi **JSON** (Untuk Menerima Data)

Library

Untuk library kita menggunakan beberapa library dari kontributor open source.



Source : <https://github.com/bywahjoe/kelasIOT>

Program

Berikut adalah program yang akan kita uji coba.

```
#include "CTBot.h"
#include "key.h"
CTBot myBot;

//Config-----
#define chatID your_chatID
String msg = your_msg;
String ssid = your_ssid;
String pass = your_pass;
String token = your_token;
//-----

void setup() {
  Serial.begin(115200);

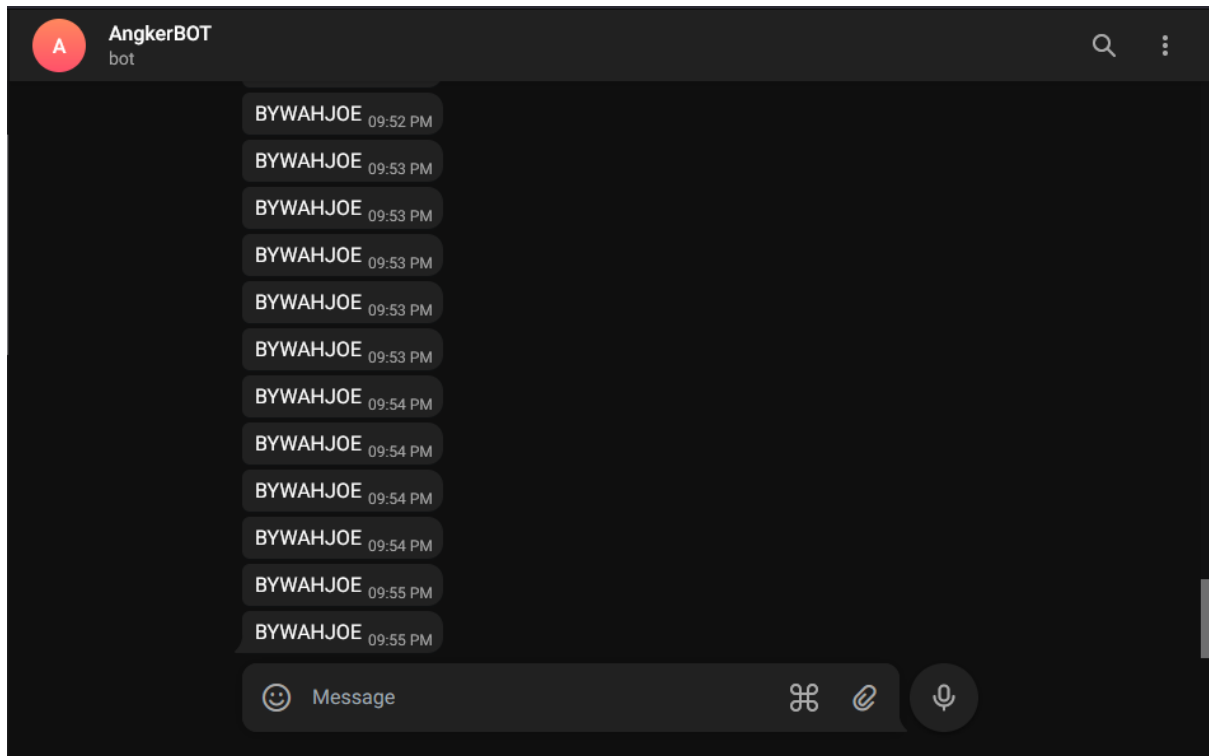
  myBot.wifiConnect(ssid, pass); //WIFI CONECT
  myBot.setTelegramToken(token); //TOKEN SETT
  delay(100);

  if (myBot.testConnection()) {
    Serial.println("\nSIGNAL OK");
  }
  else {
    Serial.println("\nSIGNAL ERR");
  }
}

void loop() {
  //Sending BYWAHJOE 10 sec
  myBot.sendMessage(chatID, msg);
  delay(10000);
}
```


Hasil Program

Berikut adalah hasil dari program diatas.



Blynk Platform

Pengenalan



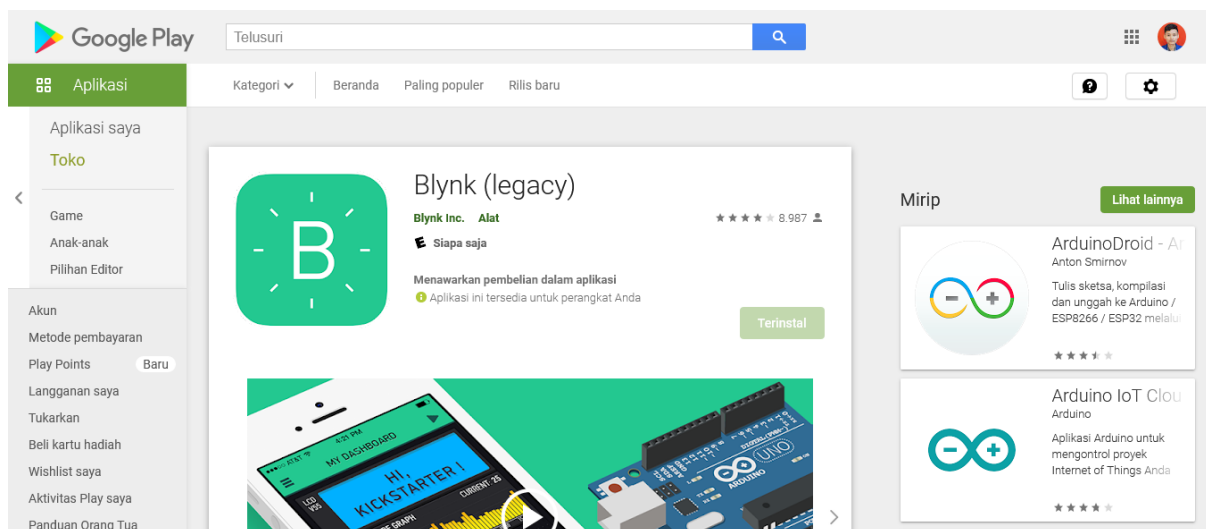
Blynk merupakan suatu platform pendukung IoT yang dapat digunakan pada smartphone. Melalui platform **Blynk** kita dapat membuat sejenis dashboard aplikasi mobile tanpa program yang rumit hanya dengan drag and drop.

Fungsi Blynk

Fungsi aplikasi Blynk ini biasanya digunakan untuk monitoring sensor pada device IoT, melakukan action tertentu.

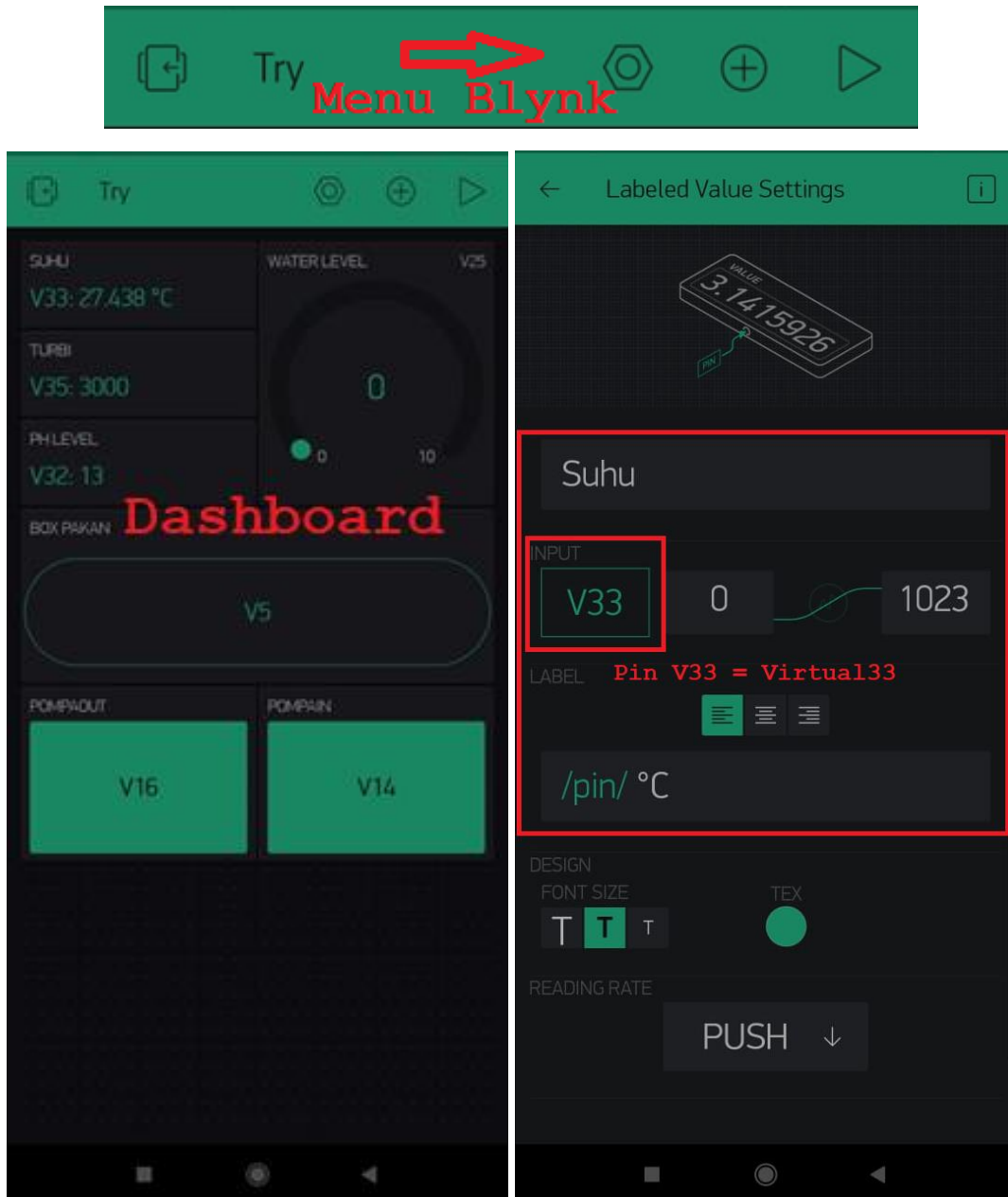
Instalasi Blynk

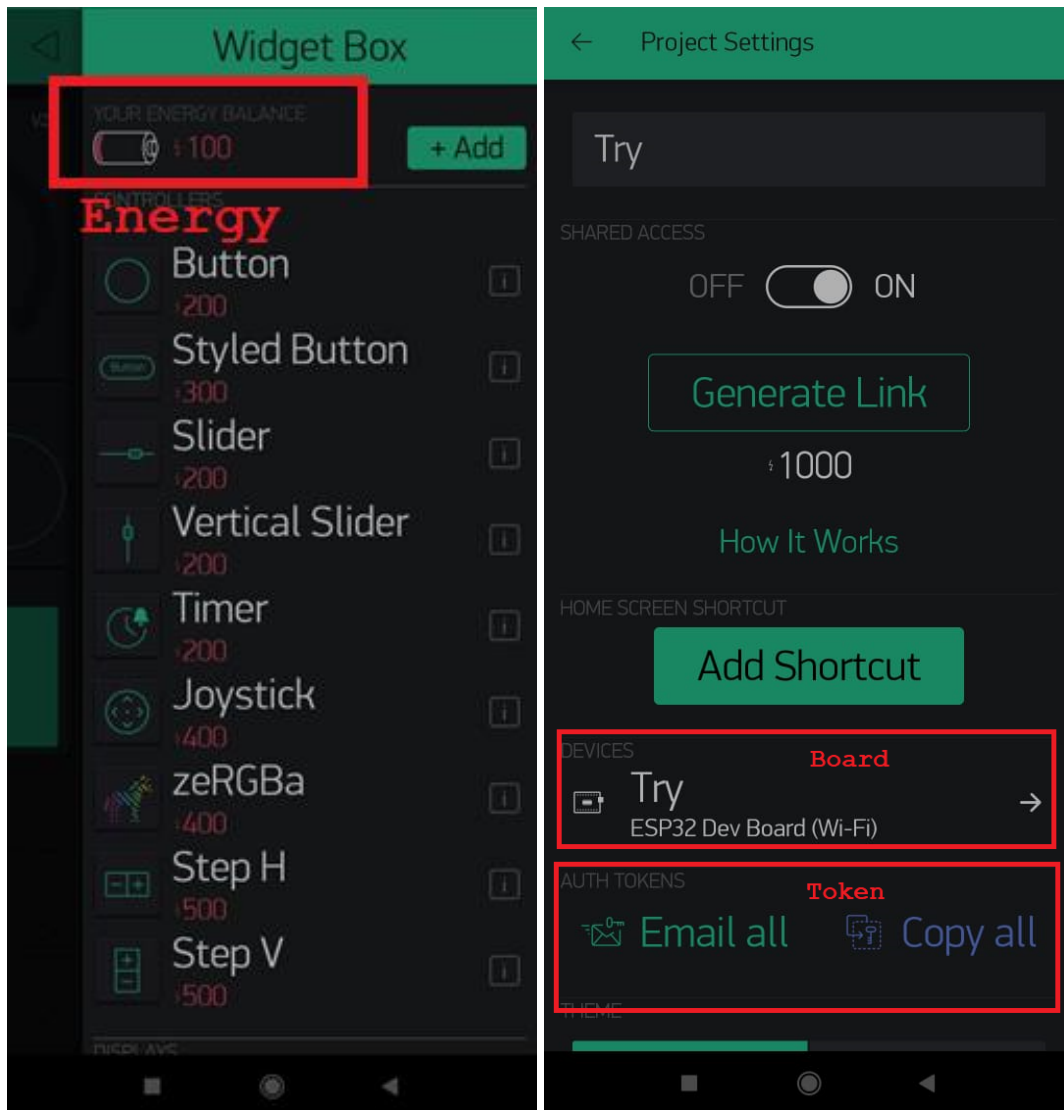
Untuk instalasi dapat di download melalui google play store untuk android.



Konfigurasi Blynk

Pada blynk terdapat banyak menu pilihan untuk drag & drop modul. Melalui panel yang telah tersedia, user dimanjakan dengan pilihannya. Namun semua modul yang telah dipilih terbatas oleh **energy**. Aplikasi blynk akan memberikan **token** untuk berkomunikasi dengan aplikasi mobile seperti mengirim data sensor.





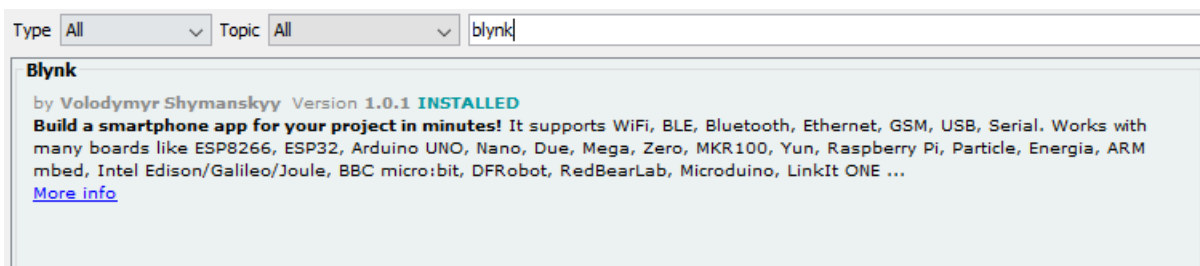
- Pada tahap ini kita telah mendapatkan **token** dan alamat **pin virtual**.

Protokol

Blynk menggunakan protokol campuran jaringan dasar seperti TCP/IP, UDP, Socket dll.

Library

Untuk library kita menggunakan beberapa library dari kontributor open source.



Source : <https://github.com/bywahjoe/kelasIOT>

Program

Berikut adalah program yang akan kita uji coba.

```
#include <BlynkSimpleEsp32.h>
#include "key.h"
#define BLYNK_PRINT Serial

//Config-----
const char* ssid = your_ssid;
const char* pass = your_pass;
const char* token = your_token;
//-----

BlynkTimer sendSensor;
void pushSensor() {

    //Push Sensor Virtual Pin 33, Random Value 20 - 39
    int sensor=random(20, 40);
    Blynk.virtualWrite(33, sensor);
}
void setup() {
    Serial.begin(115200);
    randomSeed(analogRead(0));

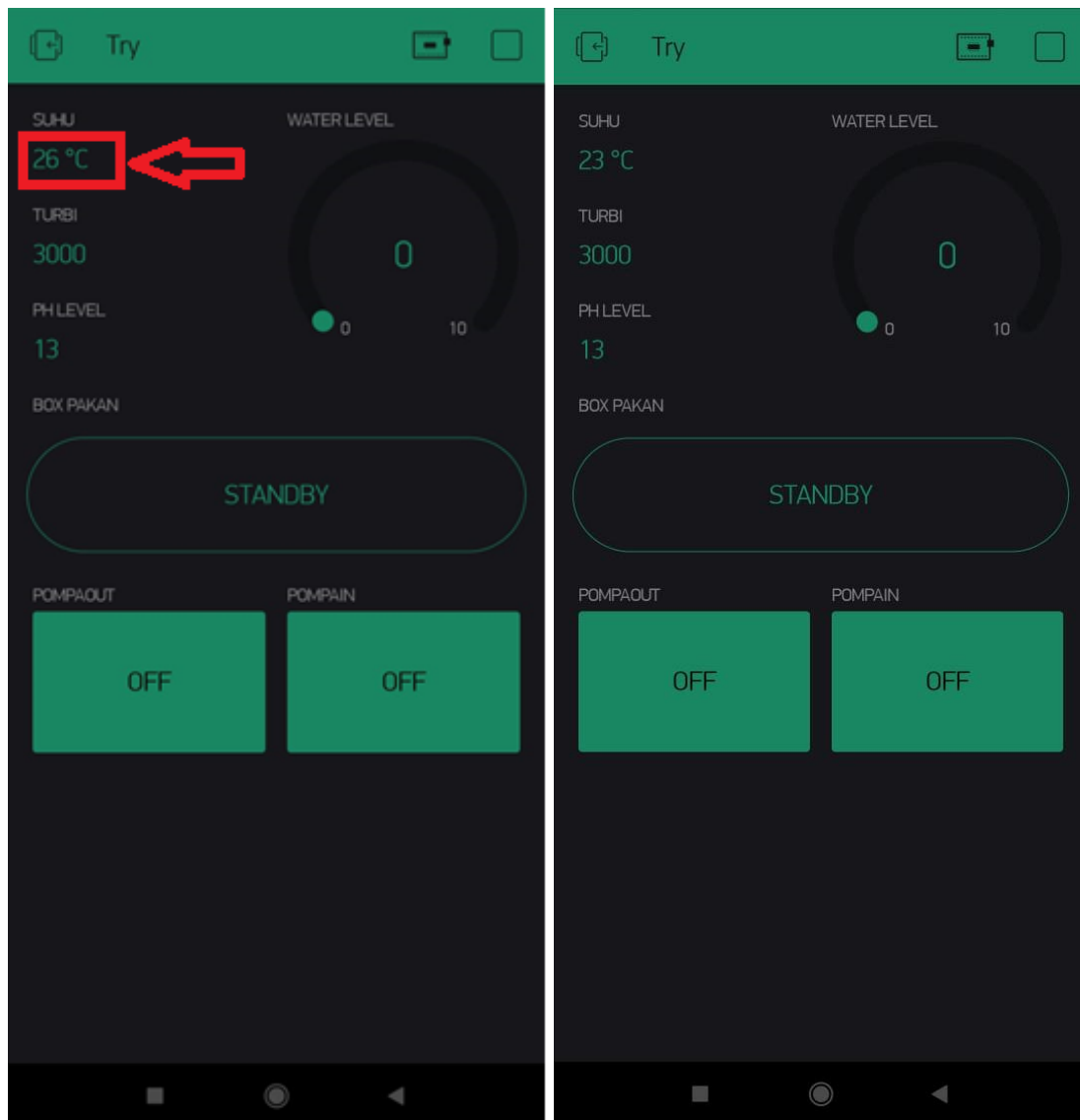
    //Blynk Config
    Blynk.begin(token, ssid, pass);
    sendSensor.setInterval(3000L, pushSensor);
}

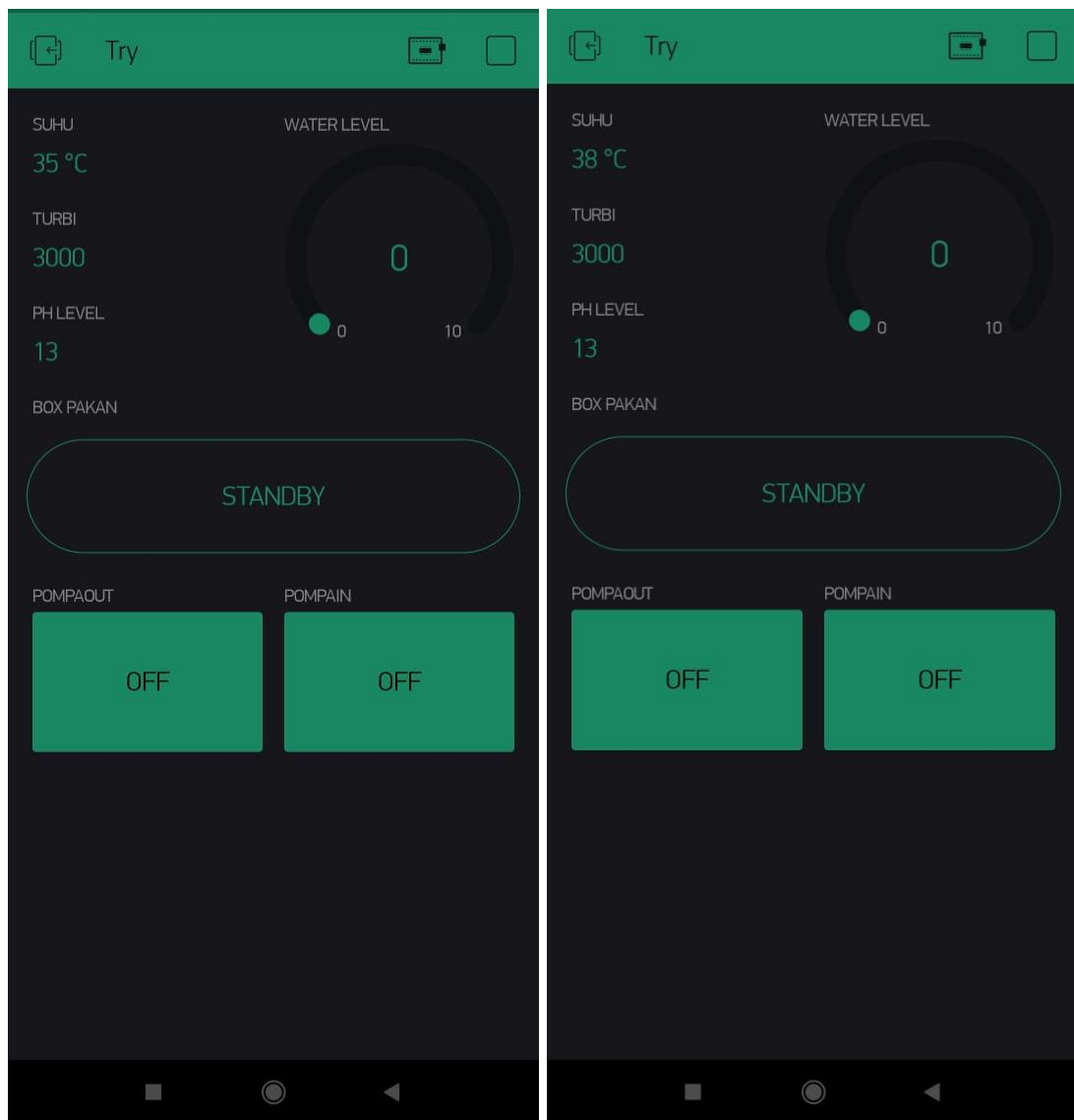
void loop() {

    //Run Blynk
    Blynk.run();
    sendSensor.run();
}
```

Hasil Program

Berikut adalah hasil dari program diatas.





HTTP


Pengenalan




Protokol **HTTP** (Hypertext Transfer Protocol) merupakan standar protokol native yang banyak digunakan dalam pembuatan **API (Application Programming Interface)**. Namun pada dasarnya menggunakan protokol **HTTP** ini membutuhkan waktu yang **lebih lama** dibandingkan **MQTT**, protokol **HTTP** sendiri masih kurang efektif untuk aplikasi IoT yang membutuhkan data realtime sangat cepat.

Debugging

Pemecahan masalah error pada protokol **HTTP** terkadang kita harus membaca **kode errornya**. Paling sering kita dengar adalah status error dengan kode **'404'** yang mempunyai keterangan bahwa situs/URL tidak valid (Not Found). Protokol bisa anda pelajari lebih lengkap disini : <https://developer.mozilla.org/en-US/docs/Web/HTTP/Status>



► Technologies► References & Guides► Feedback

Site search... (Press "/" to focus) 


Web technology for developers ► HTTP ► HTTP response status codes  [Change language](#)

Table of contents

- [Information responses](#)
- [Successful responses](#)
- [Redirection messages](#)
- [Client error responses](#)
- [Server error responses](#)
- [Browser compatibility](#)
- [See also](#)



Related Topics

[HTTP](#)

HTTP response status codes

HTTP response status codes indicate whether a specific [HTTP](#) request has been successfully completed. Responses are grouped in five classes:

- [Informational responses](#) (100 – 199)
- [Successful responses](#) (200 – 299)
- [Redirection messages](#) (300 – 399)
- [Client error responses](#) (400 – 499)
- [Server error responses](#) (500 – 599)

The below status codes are defined by [section 10 of RFC 2616](#) . You can find an updated specification in [RFC 7231](#) .

Program

Berikut adalah program yang akan kita uji coba.

```
#include <WiFi.h>
#include "key.h"
#include <HTTPClient.h>
#define URLWEB "https://iot.bywahjoe.com/post.php"

//Config-----
const char* ssid = your_ssid;
const char* pass = your_pass;
String token = your_token;
//-----

void setup() {
  Serial.begin(115200);
  randomSeed(analogRead(0));

  //Wifi Connect
  WiFi.begin(ssid, pass);
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }

  //Cek Status
  Serial.println("");
  Serial.println("WiFi connected");
  Serial.println("IP address: ");
  Serial.println(WiFi.localIP());
}

void loop() {
  // put your main code here, to run repeatedly:
  pushWeb();
  delay(10000);
}

void pushWeb() {
  //Randomize Value Sensor
  int suhu = random(20, 40);
  int humid = random(50, 90);
  int lux = random(200, 4000);
  int gas = random(1, 15);
  int air = random(1, 15);

  //Casting
  String d1 = String(suhu);
  String d2 = String(humid);
  String d3 = String(lux);
  String d4 = String(gas);
  String d5 = String(air);;
```

```
//Process HTTP
HTTPClient postWeb;

postWeb.begin(URLWEB);
postWeb.addHeader("Content-Type", "application/x-www-form-urlencoded");

String dataku = "apiKey=" + token + "&d1=" + d1 + "&d2=" + d2 + "&d3=" + d3 + "&d4=" + d4 + "&d5=" + d5;
Serial.println(dataku);

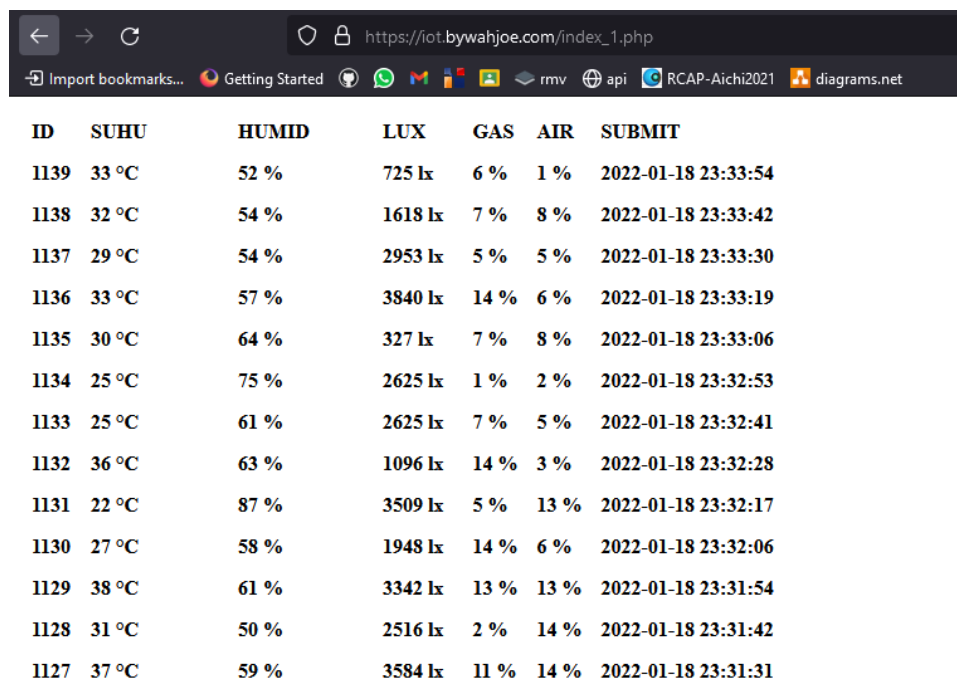
int httpResponseCode = postWeb.POST(dataku);

// UNCOMMENT TO CEK ERROR
// if (httpResponseCode>0) {
//     Serial.print("HTTP Response code: ");
//     Serial.println(httpResponseCode);
// }
// else {
//     Serial.print("Error code: ");
//     Serial.println(httpResponseCode);
// }
postWeb.end();
}
```

Hasil Program

Berikut adalah hasil dari program diatas.

- Log Data : https://iot.bywahjoe.com/index_1.php
- Dashboard : <https://iot.bywahjoe.com/>



ID	SUHU	HUMID	LUX	GAS	AIR	SUBMIT
1139	33 °C	52 %	725 lx	6 %	1 %	2022-01-18 23:33:54
1138	32 °C	54 %	1618 lx	7 %	8 %	2022-01-18 23:33:42
1137	29 °C	54 %	2953 lx	5 %	5 %	2022-01-18 23:33:30
1136	33 °C	57 %	3840 lx	14 %	6 %	2022-01-18 23:33:19
1135	30 °C	64 %	327 lx	7 %	8 %	2022-01-18 23:33:06
1134	25 °C	75 %	2625 lx	1 %	2 %	2022-01-18 23:32:53
1133	25 °C	61 %	2625 lx	7 %	5 %	2022-01-18 23:32:41
1132	36 °C	63 %	1096 lx	14 %	3 %	2022-01-18 23:32:28
1131	22 °C	87 %	3509 lx	5 %	13 %	2022-01-18 23:32:17
1130	27 °C	58 %	1948 lx	14 %	6 %	2022-01-18 23:32:06
1129	38 °C	61 %	3342 lx	13 %	13 %	2022-01-18 23:31:54
1128	31 °C	50 %	2516 lx	2 %	14 %	2022-01-18 23:31:42
1127	37 °C	59 %	3584 lx	11 %	14 %	2022-01-18 23:31:31

Source : <https://github.com/bywahjoe/kelasIOT>

TERIMA KASIH
SEMOGA BISA KETEMU KEMBALI
PADA LAIN KESEMPATAN

**Bismillah Otw Skripsi ^_^*