IoT Platform Telegram Messenger

D3 Teknologi Telekomunikasi

Fakultas Ilmu Terapan

Telkom University

Panduan Praktikum

- Panduan ini menunjukkan cara mengontrol ESP32 atau ESP8266 NodeMCU dari mana saja dengan menggunakan Telegram.
- Praktikum ini memberikan contoh untuk mengatur LED dengan hanya perlu mengirim pesan ke Bot Telegram untuk mengatur output HIGH atau LOW.
- Konfigurasi board NodeMCU yang diprogram menggunakan Arduino IDE.

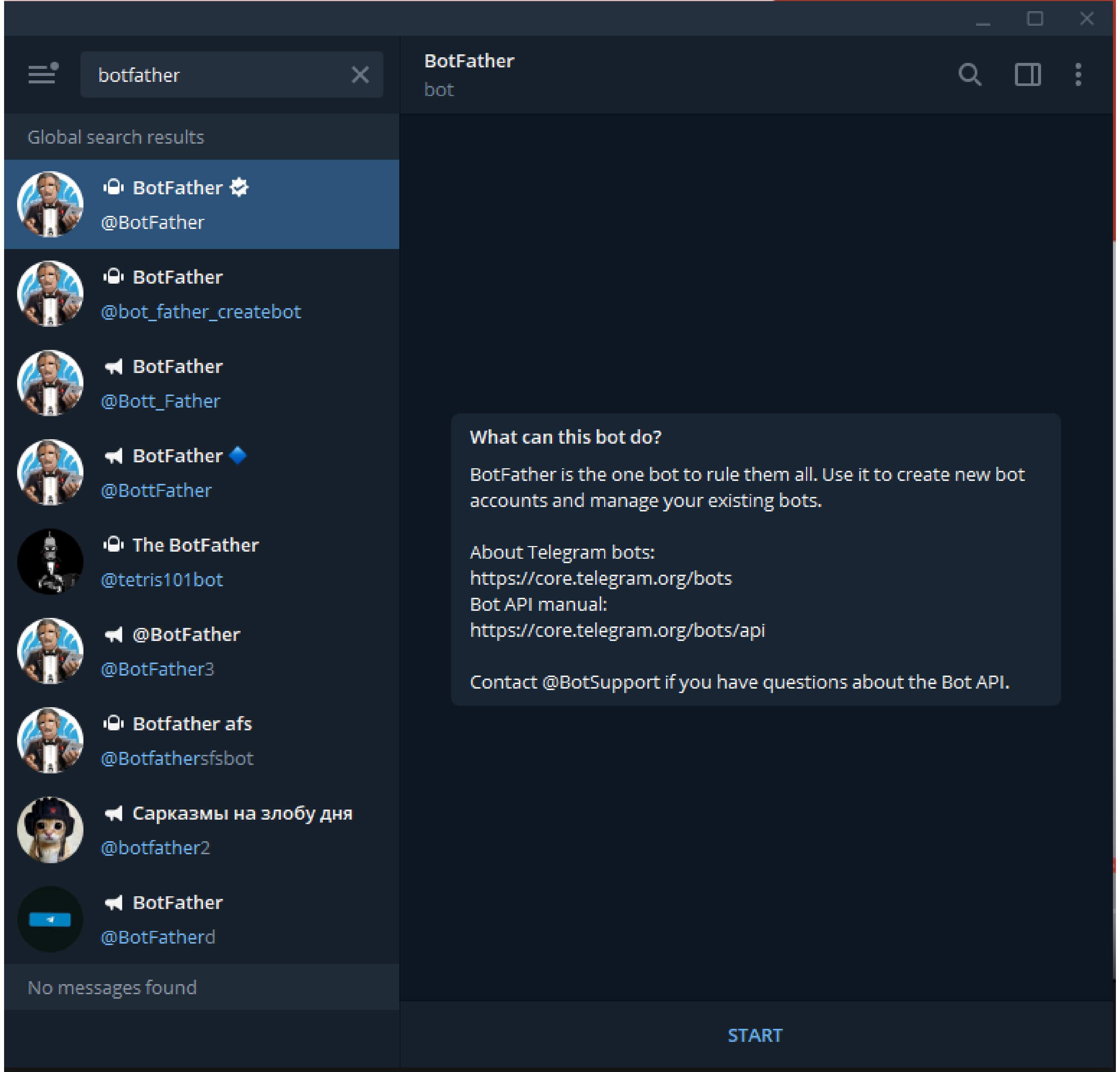
Langkah

- Setting Telegram Bot
- Konfigurasi board ESP dengan ArduinoIDE

Setting Telegram Bot

Telegram Bot

1. BotFather : Digunakan untuk mendapatkan informasi terkait dengan
   - Membuat Bot baru
   - Mendapatkan token access
2. IDBot
   - Mendapatkan ID Bot Telegram miliki kita

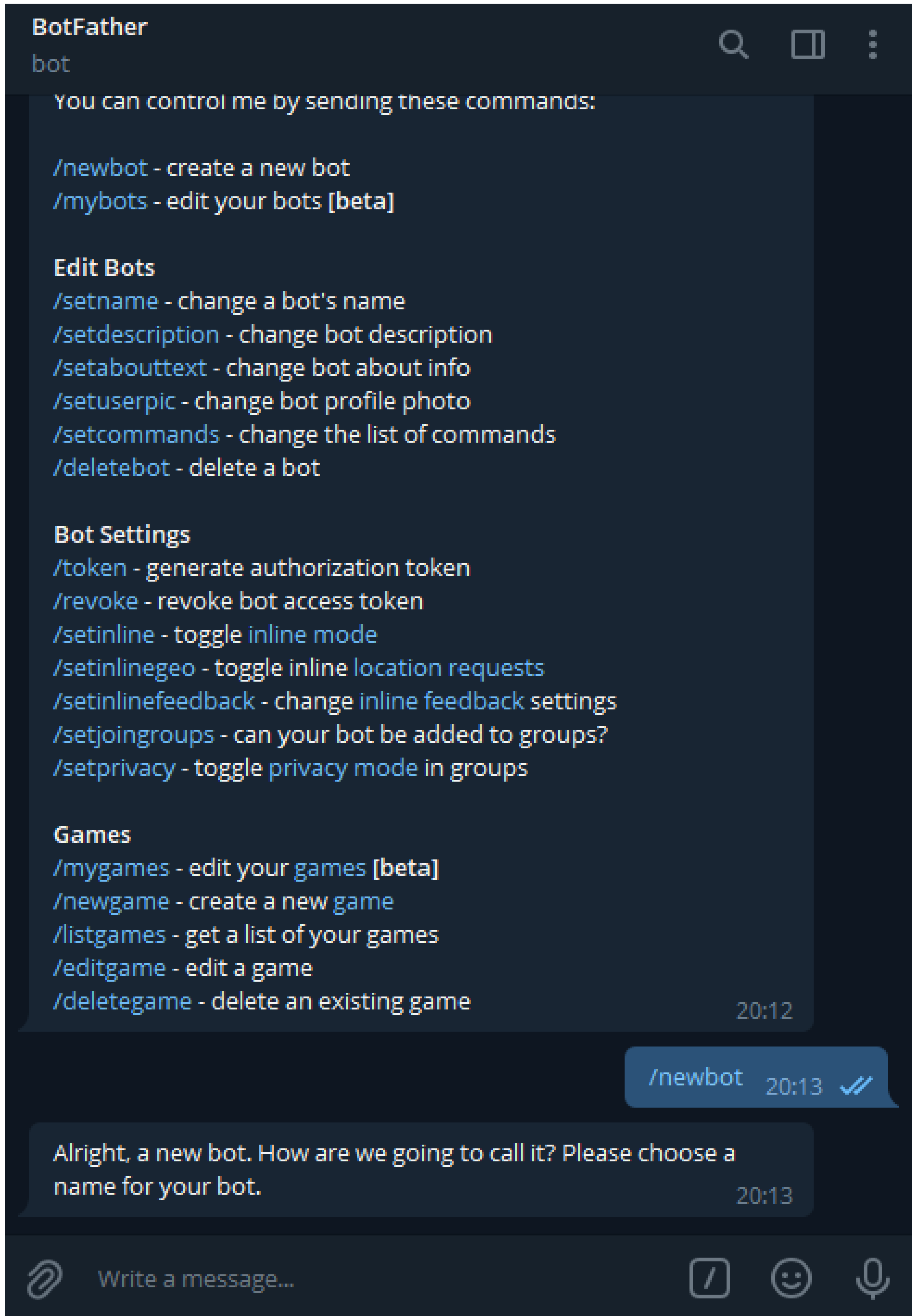


What can this bot do?

BotFather is the one bot to rule them all. Use it to create new bot accounts and manage your existing bots.

About Telegram bots:
https://core.telegram.org/bots
Bot API manual:
https://core.telegram.org/bots/api

Contact @BotSupport if you have questions about the Bot API.


BotFather
bot

You can control me by sending these commands:

/newbot - create a new bot
/mybots - edit your bots **[beta]**

Edit Bots
/setname - change a bot's name
/setdescription - change bot description
/setabouttext - change bot about info
/setuserpic - change bot profile photo
/setcommands - change the list of commands
/deletebot - delete a bot

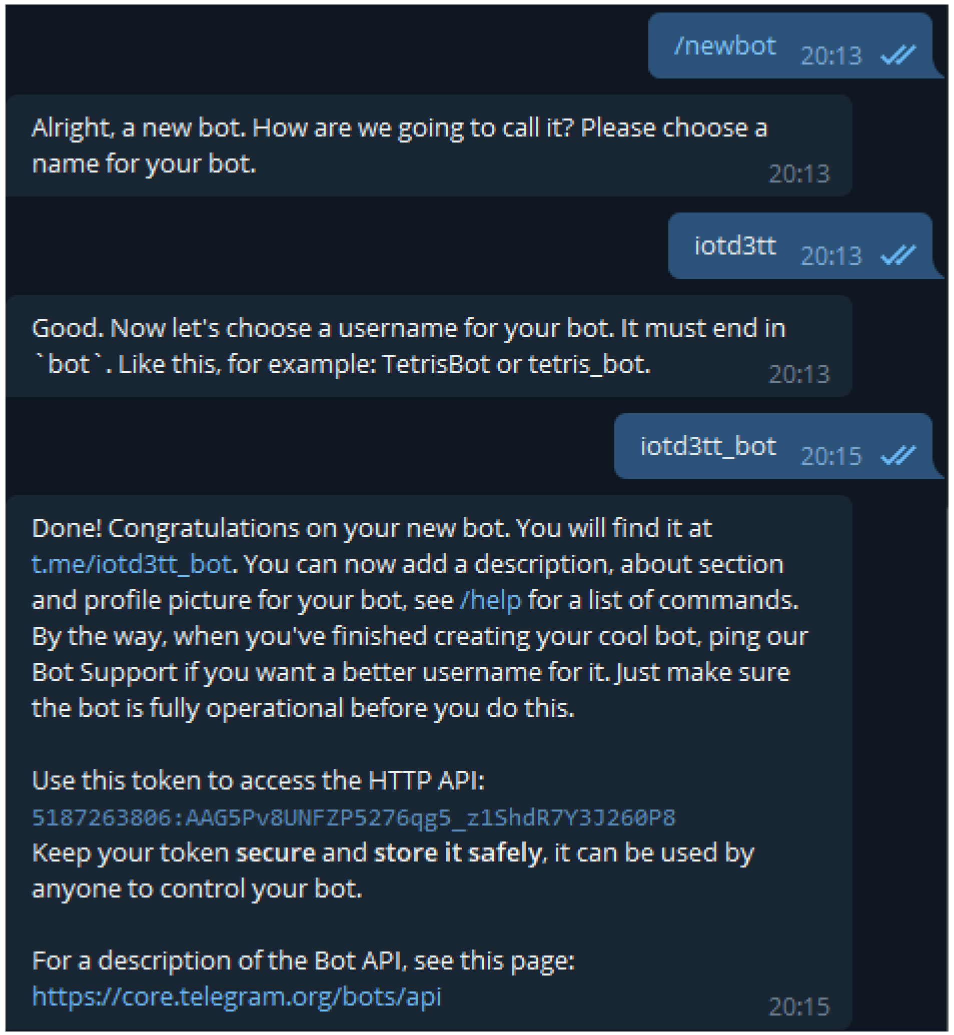
Bot Settings
/token - generate authorization token
/revoke - revoke bot access token
/setinline - toggle inline mode
/setinlinegeo - toggle inline location requests
/setinlinefeedback - change inline feedback settings
/setjoingroups - can your bot be added to groups?
/setprivacy - toggle privacy mode in groups

Games
/mygames - edit your games **[beta]**
/newgame - create a new game
/listgames - get a list of your games
/editgame - edit a game
/deletegame - delete an existing game
20:12

/newbot 20:13

Alright, a new bot. How are we going to call it? Please choose a name for your bot. 20:13

Write a message...


/newbot 20:13

Alright, a new bot. How are we going to call it? Please choose a name for your bot. 20:13

iotd3tt 20:13

Good. Now let's choose a username for your bot. It must end in `bot`. Like this, for example: TetrisBot or tetris_bot. 20:13

iotd3tt_bot 20:15

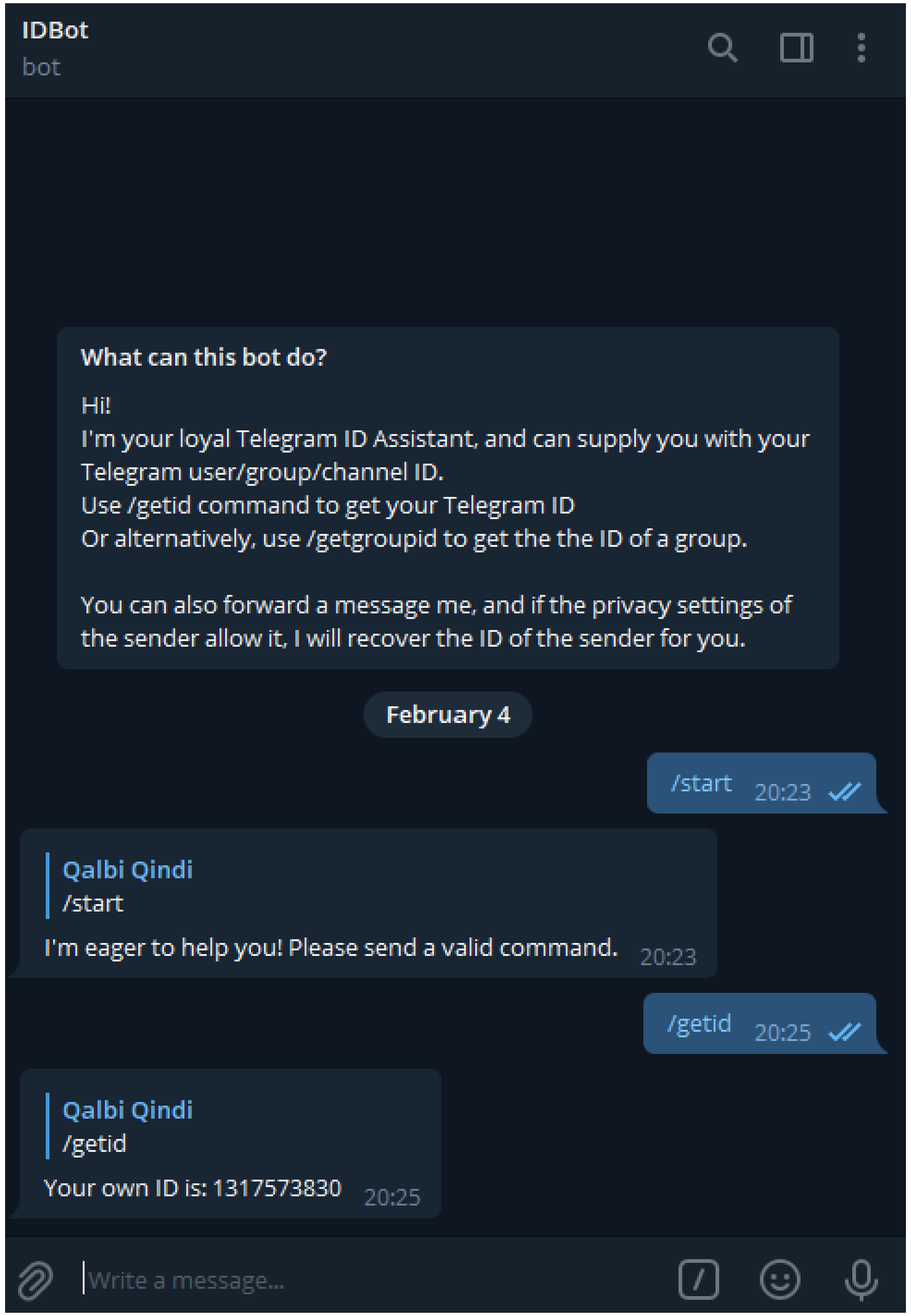
Done! Congratulations on your new bot. You will find it at t.me/iotd3tt_bot. You can now add a description, about section and profile picture for your bot, see /help for a list of commands. By the way, when you've finished creating your cool bot, ping our Bot Support if you want a better username for it. Just make sure the bot is fully operational before you do this.

Use this token to access the HTTP API:
`5187263806:AAG5Pv8UNFZP5276qg5_z1ShdR7Y3J260P8`
Keep your token **secure** and **store it safely**, it can be used by anyone to control your bot.

For a description of the Bot API, see this page:
https://core.telegram.org/bots/api 20:15

ID Telegram

- IDBot
- Simpan your own ID untuk ditempatkan pada ArduinoIDE



IDBot
bot

What can this bot do?

Hi!
I'm your loyal Telegram ID Assistant, and can supply you with your Telegram user/group/channel ID.
Use /getid command to get your Telegram ID
Or alternatively, use /getgroupid to get the the ID of a group.

You can also forward a message me, and if the privacy settings of the sender allow it, I will recover the ID of the sender for you.

February 4

/start 20:23

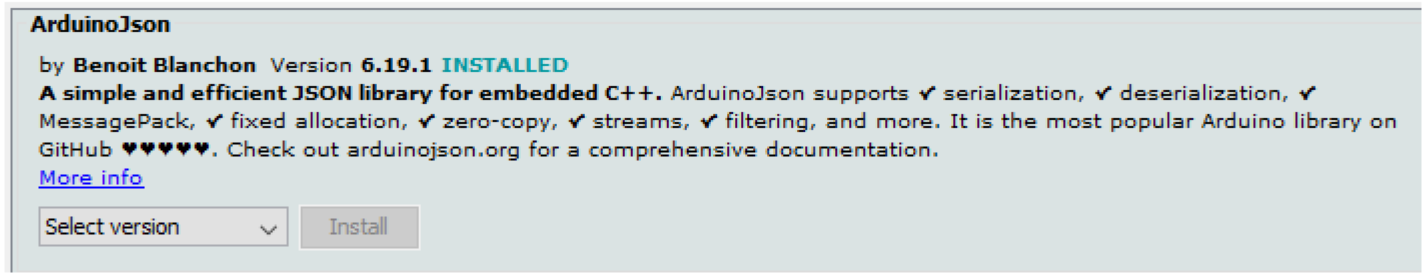
Qalbi Qindi
/start
I'm eager to help you! Please send a valid command. 20:23

/getid 20:25

Qalbi Qindi
/getid
Your own ID is: 1317573830 20:25

Konfigurasi Board ESP

Library



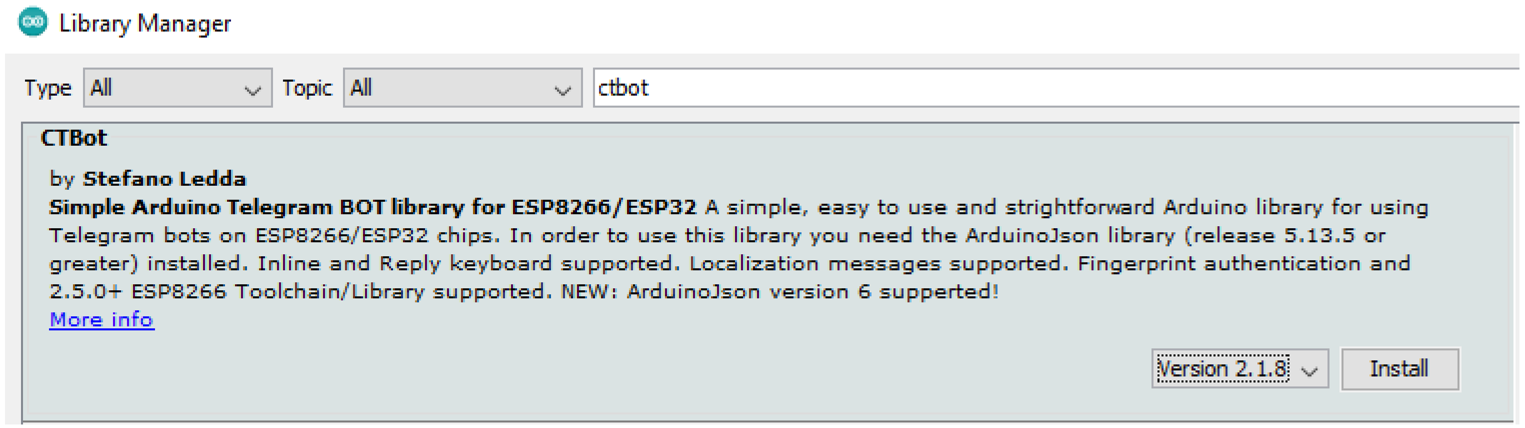
ArduinoJson

by **Benoit Blanchon** Version **6.19.1** INSTALLED

A simple and efficient JSON library for embedded C++. ArduinoJson supports ✓ serialization, ✓ deserialization, ✓ MessagePack, ✓ fixed allocation, ✓ zero-copy, ✓ streams, ✓ filtering, and more. It is the most popular Arduino library on GitHub ♥♥♥♥♥. Check out arduinojson.org for a comprehensive documentation.

More info

- Install library : Sketch → Include Library → Manage Library
- Library yang dibutuhkan :
  - CTBot
  - ArduinoJson
  - Universal Arduino Telegram Bot



ctbot

CTBot

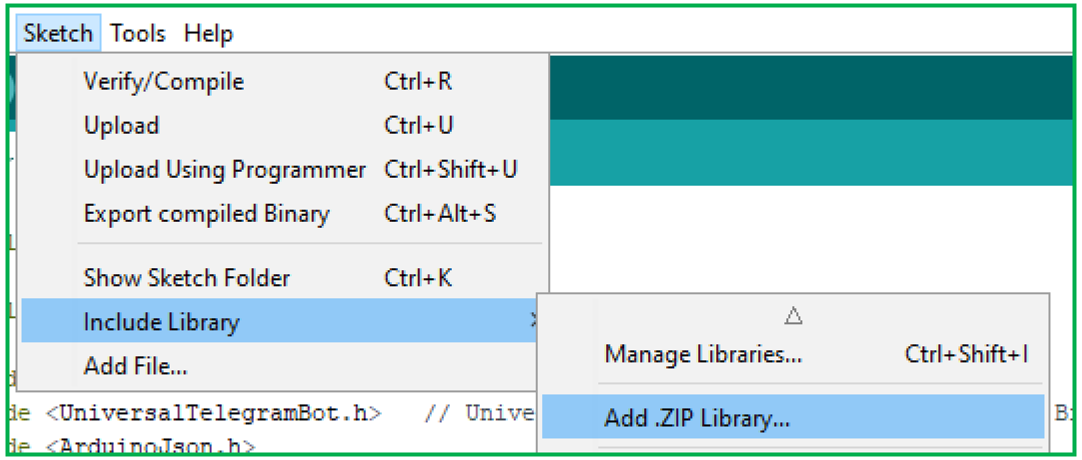
by **Stefano Ledda**

Simple Arduino Telegram BOT library for ESP8266/ESP32 A simple, easy to use and strightforward Arduino library for using Telegram bots on ESP8266/ESP32 chips. In order to use this library you need the ArduinoJson library (release 5.13.5 or greater) installed. Inline and Reply keyboard supported. Localization messages supported. Fingerprint authentication and 2.5.0+ ESP8266 Toolchain/Library supported. NEW: ArduinoJson version 6 suppoerted!

More info

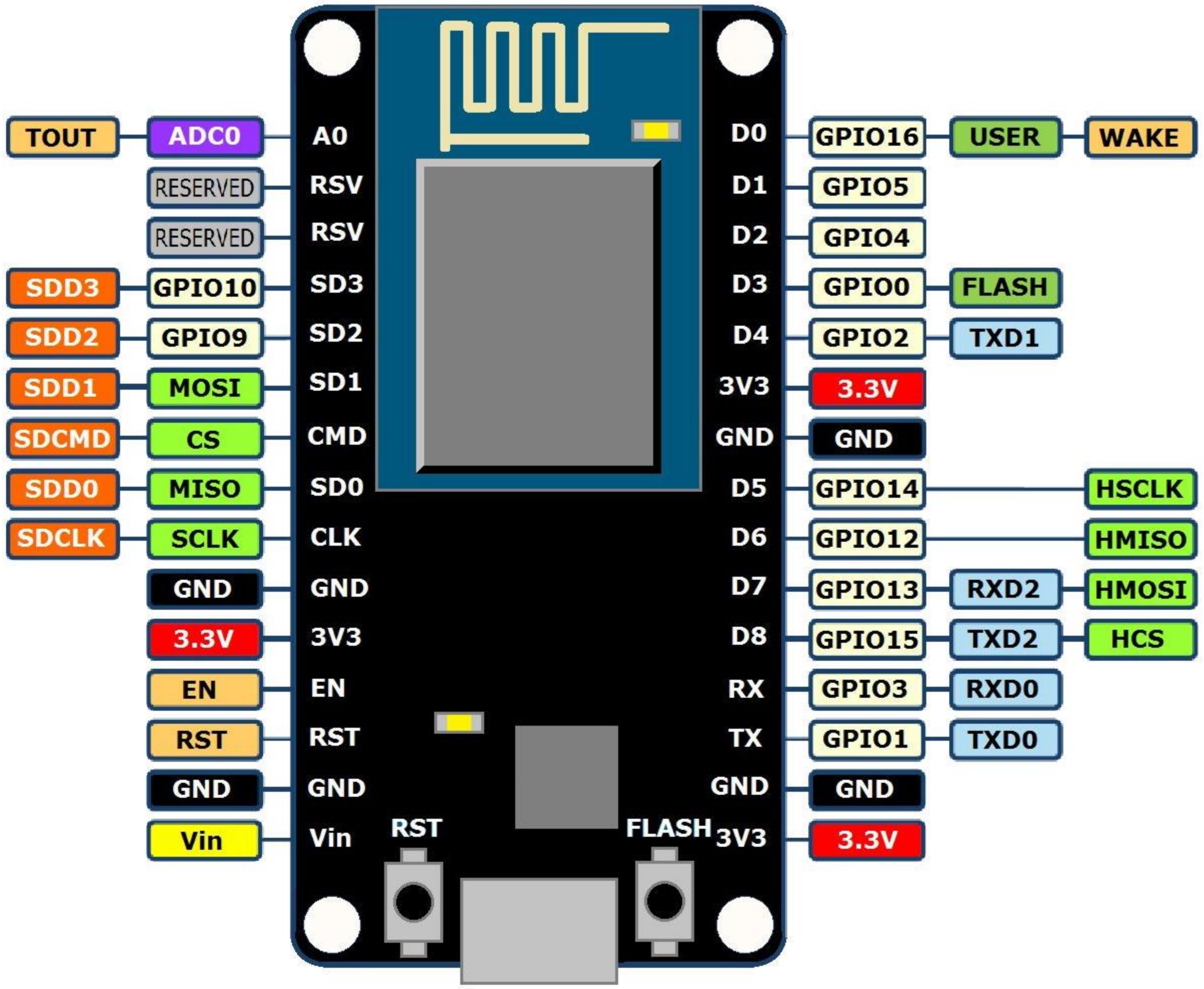
Install Library Universal Telegram Bot

- Download :
  - https://github.com/witnessmenow/Universal-Arduino-Telegram-Bot
- Tambahkan zip library yang sudah didownload pada Arduino IDE


Sketch
Tools
Help
Verify/Compile Ctrl+R
Upload Ctrl+U
Upload Using Programmer Ctrl+Shift+U
Export compiled Binary Ctrl+Alt+S
Show Sketch Folder Ctrl+K
Include Library
Add File...
Manage Libraries... Ctrl+Shift+I
Add .ZIP Library...

```
de <UniversalTelegramBot.h>    // Unive
de <ArduinoJson.h>
```

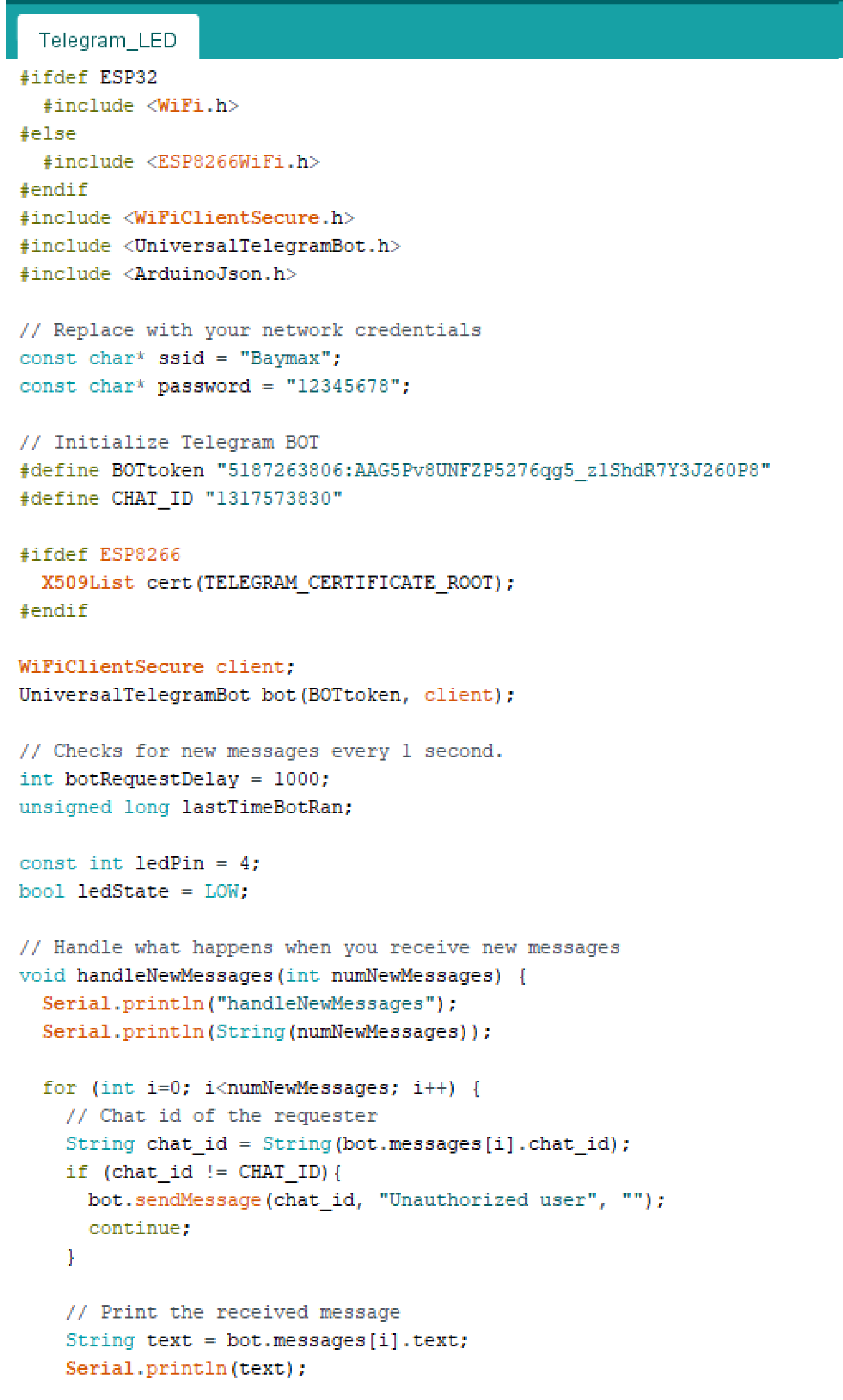
Board NodeMCU



Persiapan LED dan NodeMCU

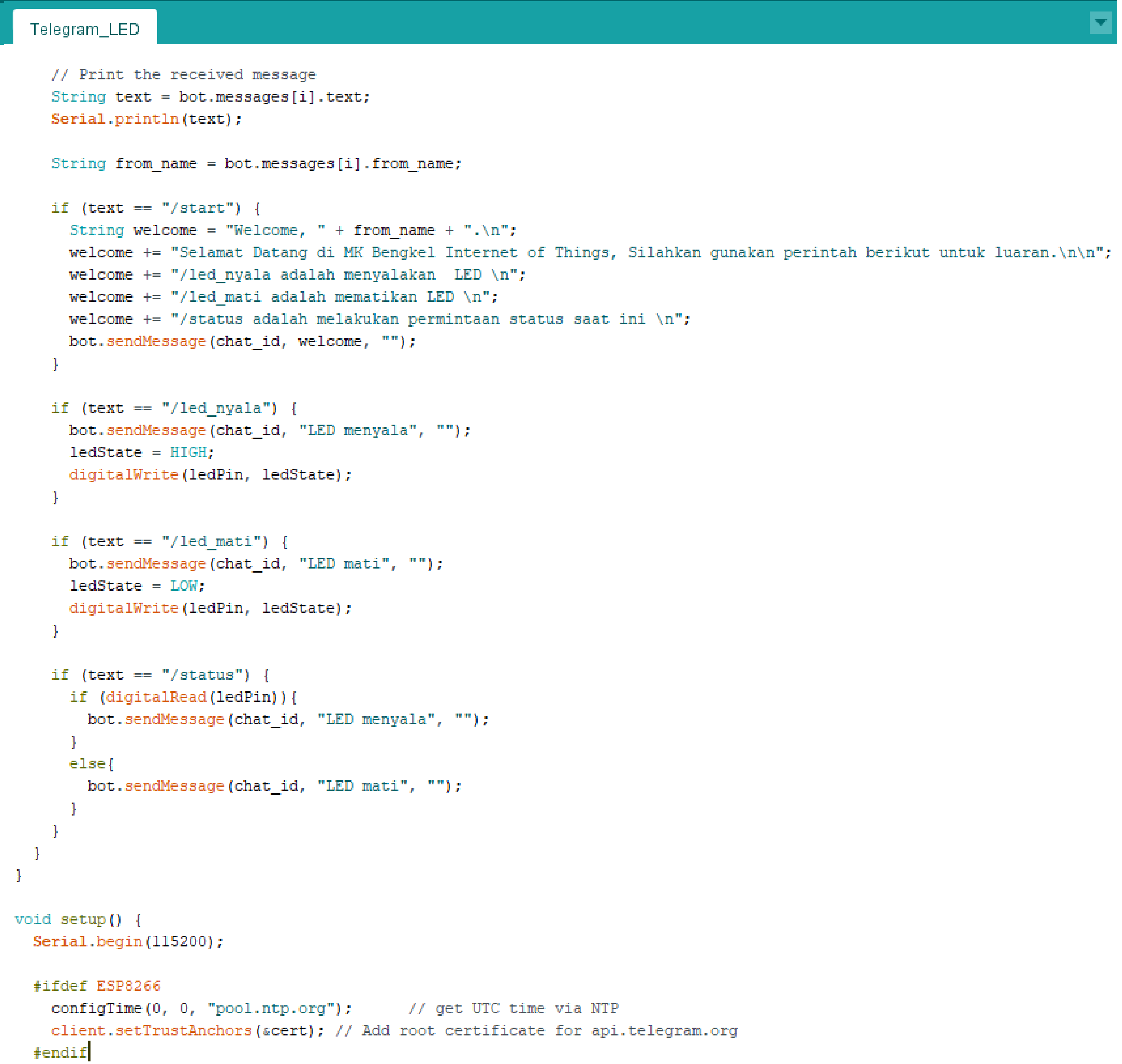
- Alat :
  - NodeMCU
  - Led (1 bh) → Kaki Pendek (Negatif), Kaki Panjang (positif)
  - Kabel jumper female (2 bh)
- Langkah Pemasangan kabel jumper
  - D2 → Kaki Panjang (Positif)
  - G → Kaki Pendek (Negatif)

Ket : D2 dan G (Ground) → lihat Board NodeMCU



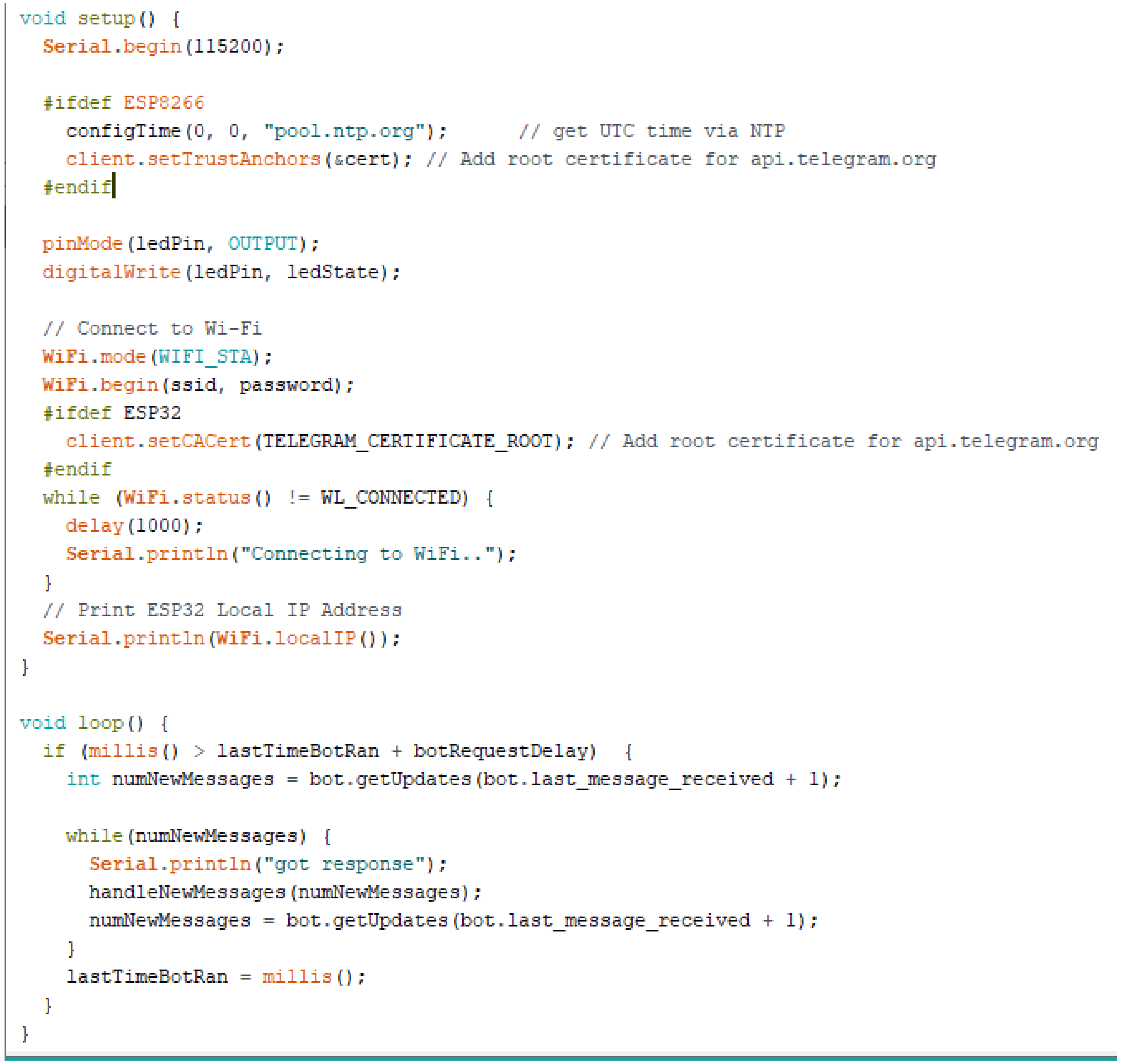
Telegram_LED

```
#ifdef ESP32
  #include <WiFi.h>
#else
  #include <ESP8266WiFi.h>
#endif
#include <WiFiClientSecure.h>
#include <UniversalTelegramBot.h>
#include <ArduinoJson.h>

// Replace with your network credentials
const char* ssid = "Baymax";
const char* password = "12345678";

// Initialize Telegram BOT
#define BOTtoken "5187263806:AAG5Pv8UNFZP5276qg5_z1ShdR7Y3J260P8"
#define CHAT_ID "1317573830"

#ifdef ESP8266
  X509List cert(TELEGRAM_CERTIFICATE_ROOT);
#endif

WiFiClientSecure client;
UniversalTelegramBot bot(BOTtoken, client);

// Checks for new messages every 1 second.
int botRequestDelay = 1000;
unsigned long lastTimeBotRan;

const int ledPin = 4;
bool ledState = LOW;

// Handle what happens when you receive new messages
void handleNewMessages(int numNewMessages) {
  Serial.println("handleNewMessages");
  Serial.println(String(numNewMessages));

  for (int i=0; i<numNewMessages; i++) {
    // Chat id of the requester
    String chat_id = String(bot.messages[i].chat_id);
    if (chat_id != CHAT_ID){
      bot.sendMessage(chat_id, "Unauthorized user", "");
      continue;
    }

    // Print the received message
    String text = bot.messages[i].text;
    Serial.println(text);
```



Telegram_LED

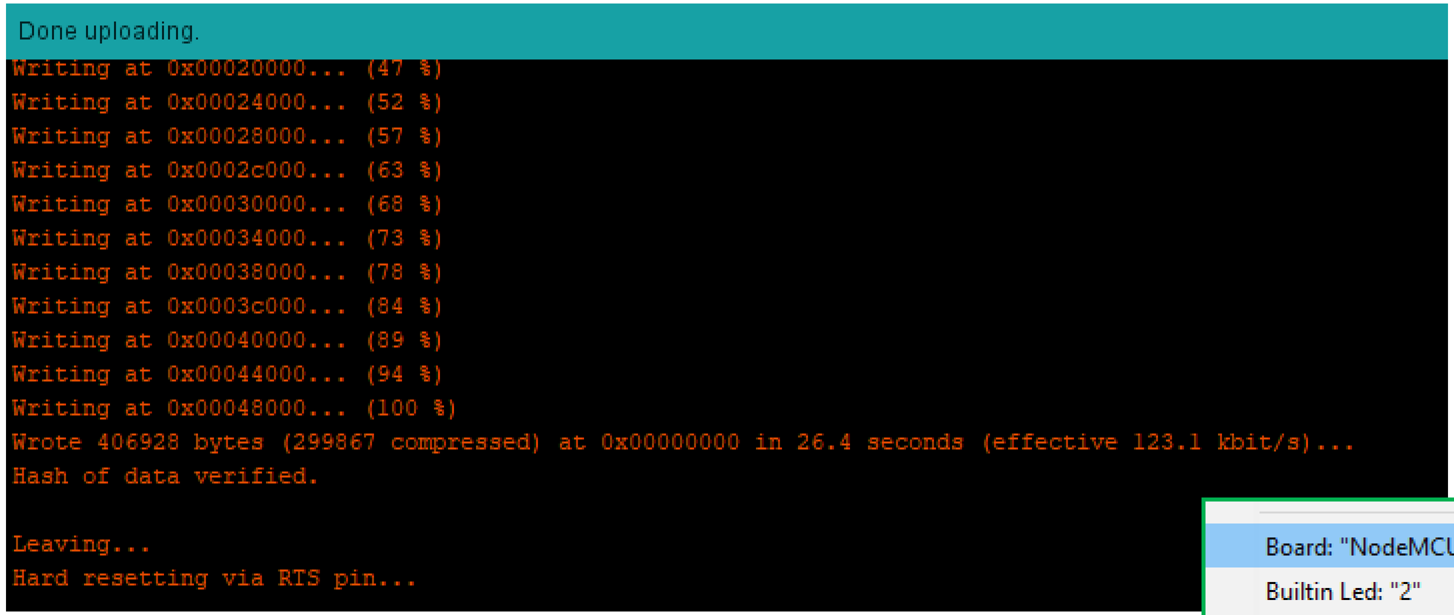
```
    // Print the received message
    String text = bot.messages[i].text;
    Serial.println(text);

    String from_name = bot.messages[i].from_name;

    if (text == "/start") {
      String welcome = "Welcome, " + from_name + ".\n";
      welcome += "Selamat Datang di MK Bengkel Internet of Things, Silahkan gunakan perintah berikut untuk luaran.\n\n";
      welcome += "/led_nyala adalah menyalakan  LED \n";
      welcome += "/led_mati adalah mematikan LED \n";
      welcome += "/status adalah melakukan permintaan status saat ini \n";
      bot.sendMessage(chat_id, welcome, "");
    }

    if (text == "/led_nyala") {
      bot.sendMessage(chat_id, "LED menyala", "");
      ledState = HIGH;
      digitalWrite(ledPin, ledState);
    }

    if (text == "/led_mati") {
      bot.sendMessage(chat_id, "LED mati", "");
      ledState = LOW;
      digitalWrite(ledPin, ledState);
    }

    if (text == "/status") {
      if (digitalRead(ledPin)){
        bot.sendMessage(chat_id, "LED menyala", "");
      }
      else{
        bot.sendMessage(chat_id, "LED mati", "");
      }
    }
  }
}

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

  #ifdef ESP8266
    configTime(0, 0, "pool.ntp.org");      // get UTC time via NTP
    client.setTrustAnchors(&cert); // Add root certificate for api.telegram.org
  #endif
```



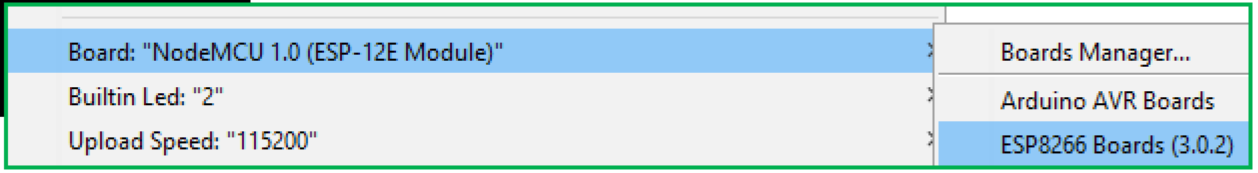
```
void setup() {
  Serial.begin(115200);

  #ifdef ESP8266
    configTime(0, 0, "pool.ntp.org");      // get UTC time via NTP
    client.setTrustAnchors(&cert); // Add root certificate for api.telegram.org
  #endif

  pinMode(ledPin, OUTPUT);
  digitalWrite(ledPin, ledState);

  // Connect to Wi-Fi
  WiFi.mode(WIFI_STA);
  WiFi.begin(ssid, password);
  #ifdef ESP32
    client.setCACert(TELEGRAM_CERTIFICATE_ROOT); // Add root certificate for api.telegram.org
  #endif
  while (WiFi.status() != WL_CONNECTED) {
    delay(1000);
    Serial.println("Connecting to WiFi..");
  }
  // Print ESP32 Local IP Address
  Serial.println(WiFi.localIP());
}

void loop() {
  if (millis() > lastTimeBotRan + botRequestDelay)  {
    int numNewMessages = bot.getUpdates(bot.last_message_received + 1);

    while(numNewMessages) {
      Serial.println("got response");
      handleNewMessages(numNewMessages);
      numNewMessages = bot.getUpdates(bot.last_message_received + 1);
    }
    lastTimeBotRan = millis();
  }
}
```

Upload

- Pastikan tidak ada error
- Cek Port yang digunakan
- “ESP8266 Board” menggunakan versi yang terbaru.



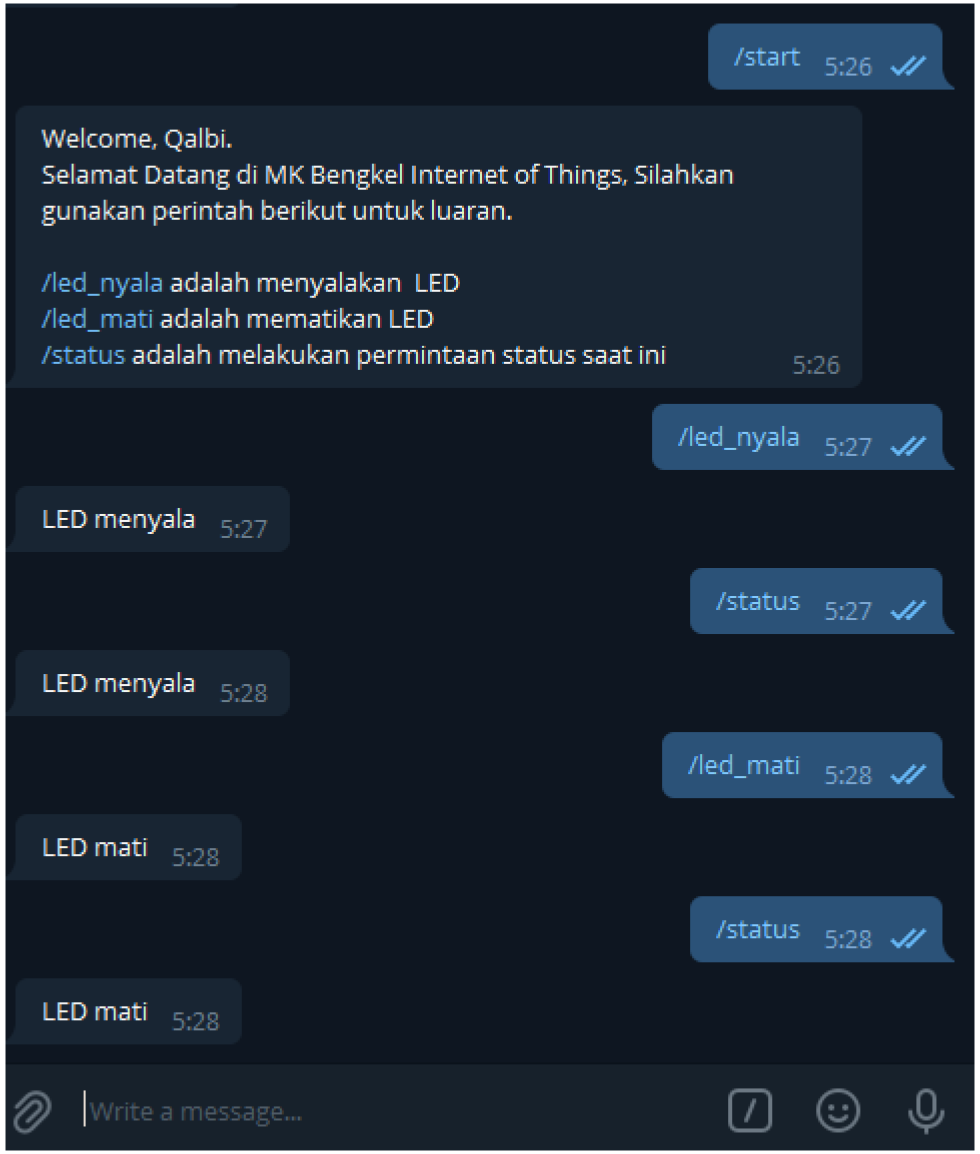
```
Done uploading.
Writing at 0x00020000... (47 %)
Writing at 0x00024000... (52 %)
Writing at 0x00028000... (57 %)
Writing at 0x0002c000... (63 %)
Writing at 0x00030000... (68 %)
Writing at 0x00034000... (73 %)
Writing at 0x00038000... (78 %)
Writing at 0x0003c000... (84 %)
Writing at 0x00040000... (89 %)
Writing at 0x00044000... (94 %)
Writing at 0x00048000... (100 %)
Wrote 406928 bytes (299867 compressed) at 0x00000000 in 26.4 seconds (effective 123.1 kbit/s)...
Hash of data verified.

Leaving...
Hard resetting via RTS pin...
```



Board: "NodeMCU 1.0 (ESP-12E Module)"
Builtin Led: "2"
Upload Speed: "115200"

Hasil

- /start digunakan untuk mendapatkan informasi terkait dengan perintah yang akan digunakan
- /led_nyala merupakan perintah kepada board ESP8266 untuk menyalakan LED dan mendapatkan respon dari board dengan keterangan “LED menyala”
- /led_mati merupakan perintah kepada board ESP8266 untuk menyalakan LED dan mendapatkan respon dari board dengan keterangan “LED mati”
- /status digunakan untuk mendapatkan informasi saat ini, apakah LED dalam keadaan mati atau menyala.



/start 5:26

Welcome, Qalbi.
Selamat Datang di MK Bengkel Internet of Things, Silahkan gunakan perintah berikut untuk luaran.

/led_nyala adalah menyalakan LED
/led_mati adalah mematikan LED
/status adalah melakukan permintaan status saat ini 5:26

/led_nyala 5:27

LED menyala 5:27

/status 5:27

LED menyala 5:28

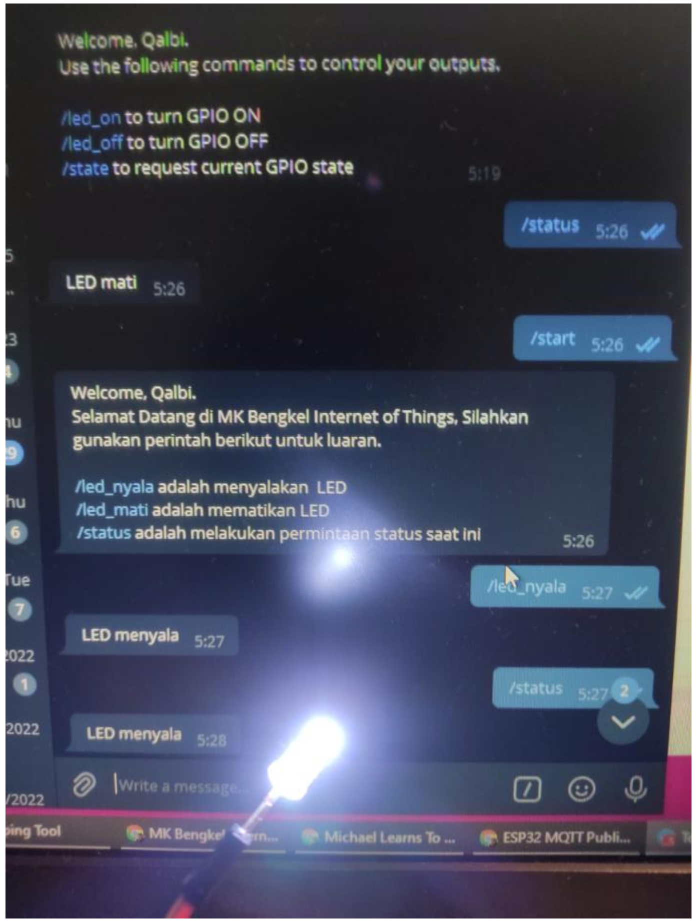
/led_mati 5:28

LED mati 5:28

/status 5:28

LED mati 5:28

Kondisi Kontrol LED Telegram



Welcome, Qalbi.
Use the following commands to control your outputs.

/led_on to turn GPIO ON
/led_off to turn GPIO OFF
/state to request current GPIO state 5:19

/status 5:26

LED mati 5:26

/start 5:26

Welcome, Qalbi.
Selamat Datang di MK Bengkel Internet of Things, Silahkan gunakan perintah berikut untuk luaran.

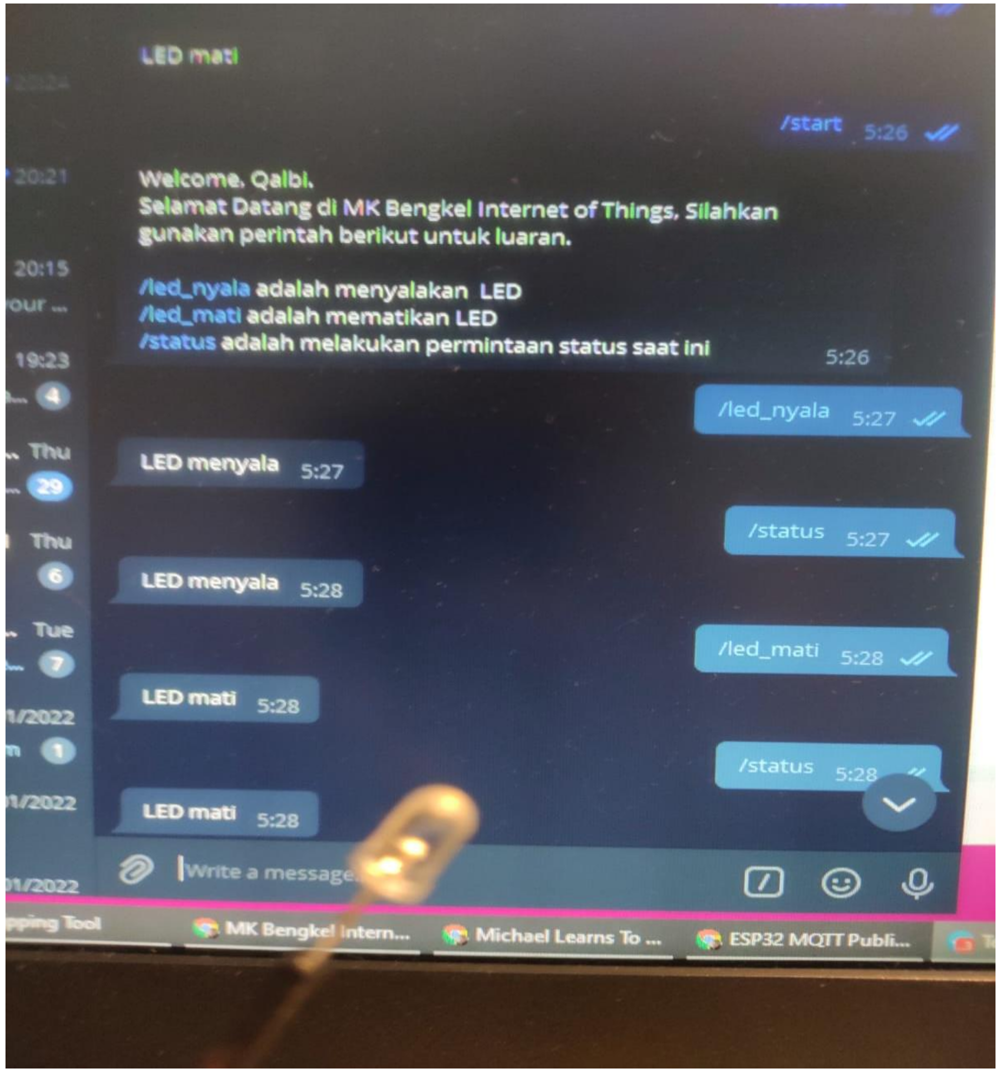
/led_nyala adalah menyalakan LED
/led_mati adalah mematikan LED
/status adalah melakukan permintaan status saat ini 5:26

/led_nyala 5:27

LED menyala 5:27

/status 5:27

LED menyala 5:28



LED mati

/start 5:26

Welcome, Qalbi.
Selamat Datang di MK Bengkel Internet of Things, Silahkan gunakan perintah berikut untuk luaran.

/led_nyala adalah menyalakan LED
/led_mati adalah mematikan LED
/status adalah melakukan permintaan status saat ini 5:26

/led_nyala 5:27

LED menyala 5:27

/status 5:27

LED menyala 5:28

/led_mati 5:28

LED mati 5:28

/status 5:28

LED mati 5:28

Studi Kasus

- Karena tidak menggunakan ESP32, hilangkan library ESP32 pada script.
- Tambahkan 1 LED dengan warna yang berbeda pada aplikasi tersebut.

Telkom University

Terima Kasih