

CLOUD COMPUTING DAN INTERNET OF THINGS

Modul VII Praktikum Sistem Tertanam



INSTITUT TEKNOLOGI SUMATERA
LAMPUNG SELATAN
2022

Daftar Isi

1. Persiapan Praktikum	2
2. Pendahuluan	6
a. Cloud Computing	6
b. Firebase	7
c. Internet of Things	8
d. Telegram	9
3. Percobaan	10
a. Firebase Projects	10
b. Simple BOT	20
4. Challenge	27
a. Simple Read Write (Firebase)	27
b. Subscriber Mode (Telegram)	27
Referensi	28

1. Persiapan Praktikum

a. Alat & Bahan

- ESP32
- Resistor Variabel
- Arduino IDE
- Jumper
- 1 Buah Sensor bebas
- WiFi Module (ESP01) -> untuk pengguna Arduino Uno

b. Catatan Tambahan

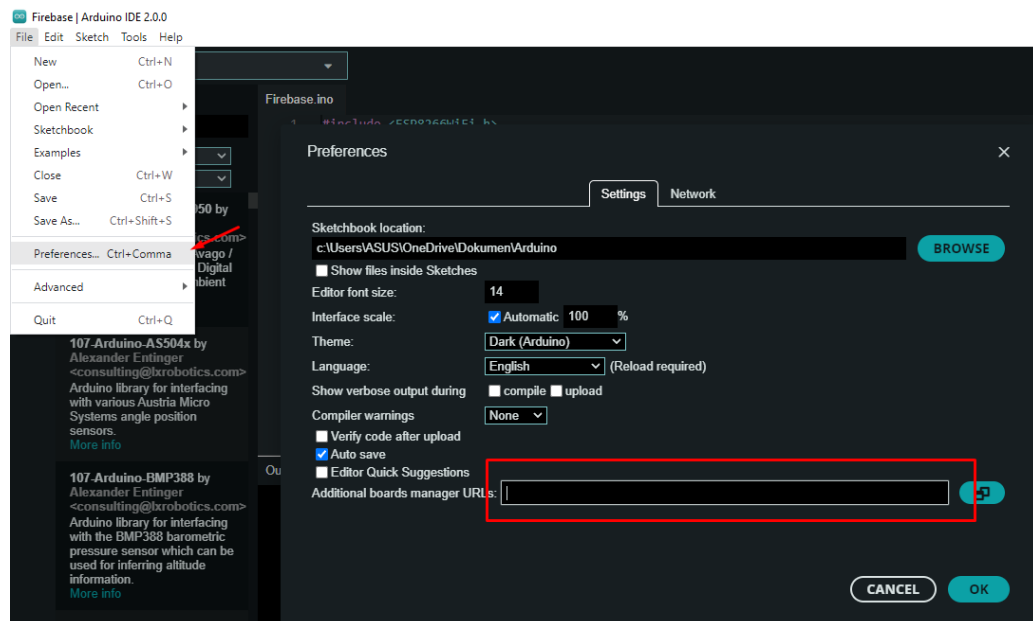
Diharapkan telah menginstall seluruh *library* pada modul ini sebelum praktikum dilaksanakan. Untuk pengguna Arduino harap menginstall **board ESP8266** terlebih dahulu.

c. Informasi Tambahan Khusus Pengguna Arduino

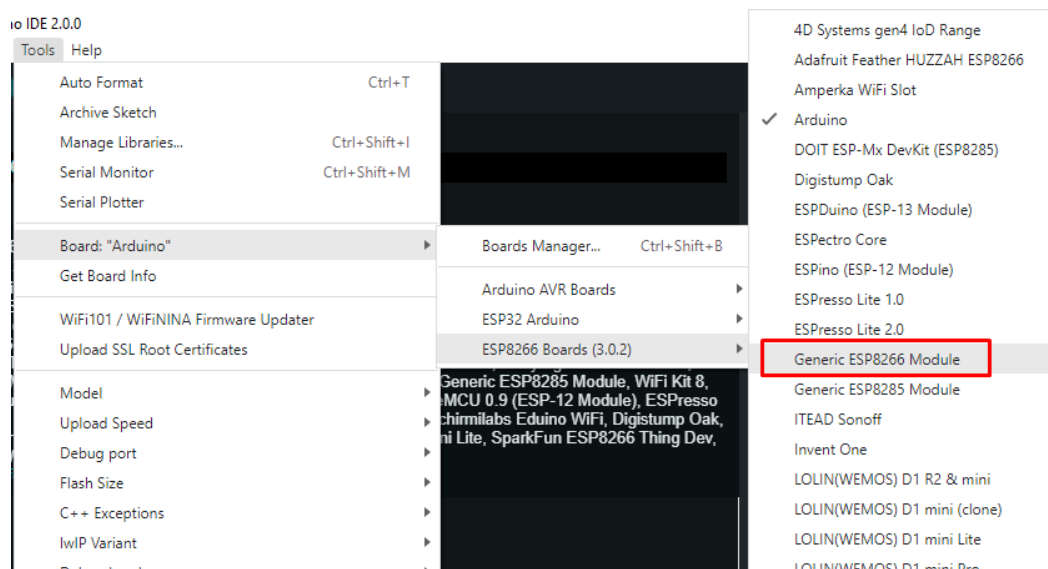
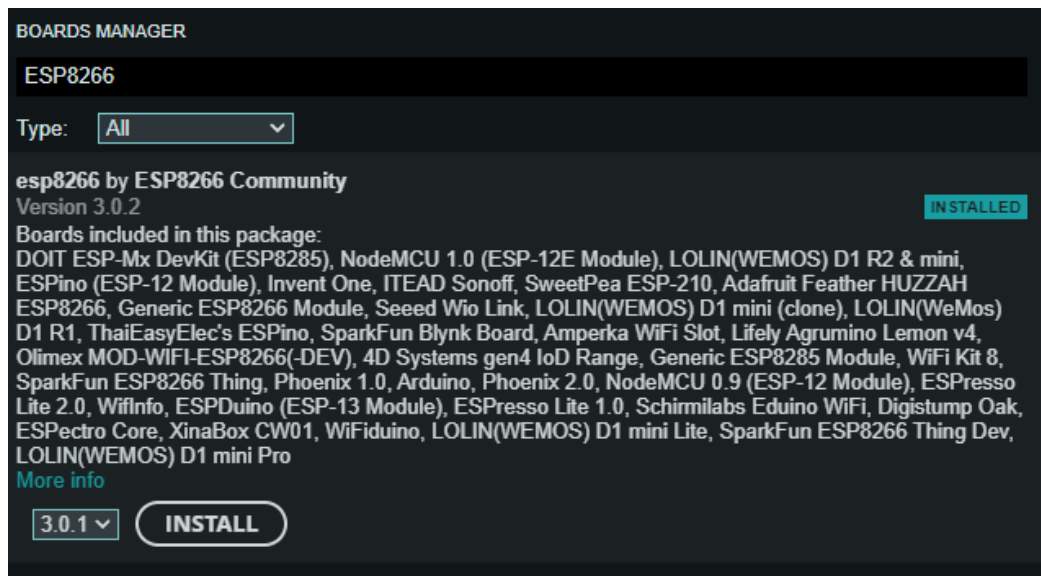
1. Install Board ESP8266

Atur preferences board manager menjadi

https://arduino.esp8266.com/stable/package_esp8266com_index.json

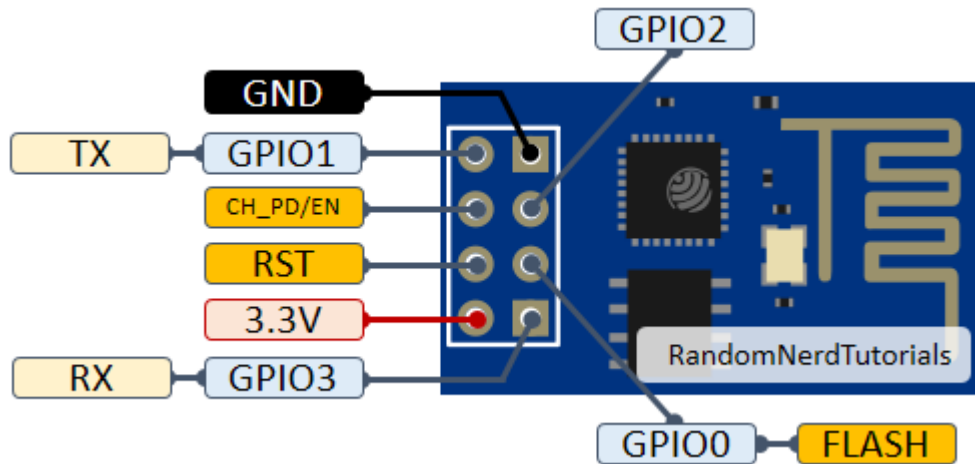


Install Board Manager ESP8266 seperti pada gambar dibawah ini. Jika sudah berhasil terinstall pilih board ESP8266.

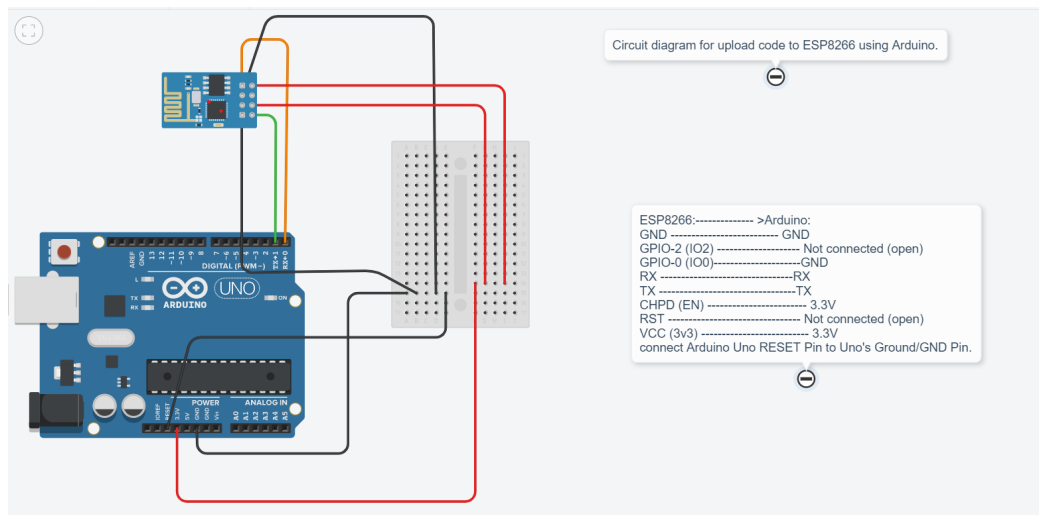


2. Cara Upload Code

Sebelum upload code ada beberapa hal yang harus diperhatikan yaitu penggunaan pinGPIO-0 (FLASH) WiFi Module dan juga RST Arduino. Perlu dicatat pinout WiFi Module secara default adalah sebagai berikut:

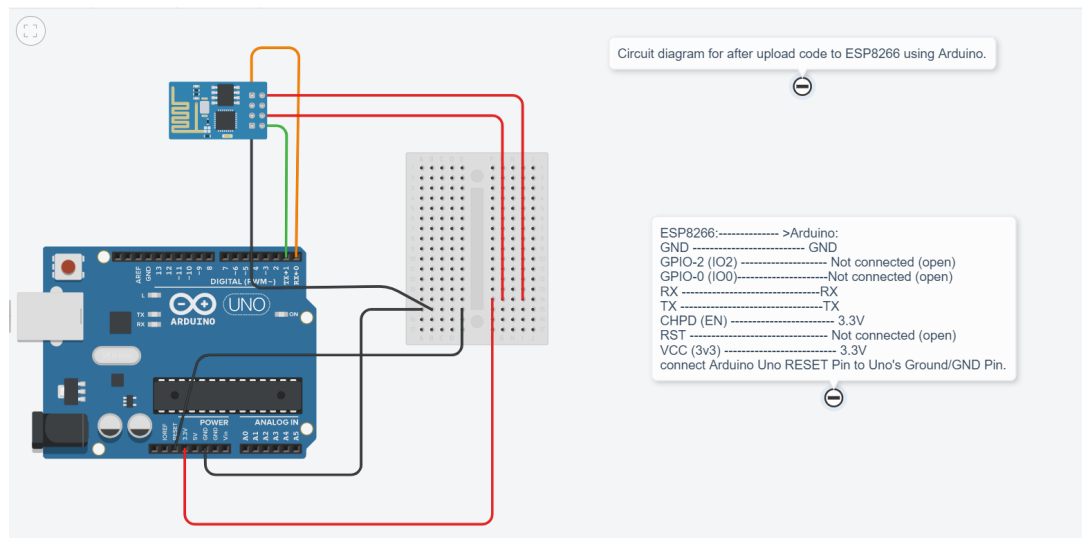


Saat upload code atur wiring Arduino dan WiFi module menjadi sebagai berikut:



ESP8266:----- >Arduino:
GND ----- GND
GPIO-2 (IO2) ----- Not connected (open)
GPIO-0 (IO0)-----GND
RX -----RX
TX -----TX
CHPD (EN) ----- 3.3V
RST ----- Not connected (open)
VCC (3v3) ----- 3.3V
connect Arduino Uno RESET Pin to Uno's Ground/GND Pin.

Jika sudah terupload, maka lepaskan pin RST dari GND dan pin GPIO-0 dari GND.



ESP8266:----- >Arduino:
GND ----- GND
GPIO-2 (IO2) ----- Not connected (open)
GPIO-0 (IO0)-----Not connected (open)
RX -----RX
TX -----TX
CHPD (EN) ----- 3.3V
RST ----- Not connected (open)
VCC (3v3) ----- 3.3V
disconnect Arduino Uno RESET Pin to Uno's Ground/GND Pin.

2. Pendahuluan

a. Cloud Computing

Cloud Computing, yang secara sederhana dapat diterjemahkan sebagai “komputasi awan”, adalah penggunaan teknologi komputasi berbasis internet. Ilustrasi *Cloud* sering digunakan untuk mewakili Internet saat membuat diagram untuk menjelaskan Internet. Mungkin dari sinilah istilah *cloud computing* berasal. *Cloud computing* menyediakan layanan berupa *software*, akses data, dan penyimpanan data. Pengguna tidak perlu tahu di mana penyedia layanan berada secara fisik atau bagaimana layanan itu bekerja [1].



Gambar 1. Ilustrasi Cloud Computing (<https://indonesiancloud.com/>)

Awan sebagai gambaran Internet, dan Anda tidak perlu memberitahu pengguna di mana letaknya. Penting bahwa pengguna Anda dapat terhubung ke Internet. Baik melalui jaringan telepon, jaringan kabel, jaringan hotspot, jaringan seluler, atau warnet, yang terpenting adalah memiliki koneksi internet yang cepat dan gratis (jika memungkinkan). Komputasi awan, atau komputasi awan, adalah tren baru di bidang komputasi

terdistribusi yang memungkinkan berbagai pihak untuk mengembangkan aplikasi dan layanan berbasis SOA (Service Oriented Architecture) melalui jaringan Internet. Berbagai kalangan dapat memanfaatkan layanan cloud computing ini sebagai solusi teknologi sekaligus mendapatkan keuntungan ekonomis darinya [1].

b. Firebase

Firebase adalah *platform* untuk aplikasi *realtime*. Saat data berubah, aplikasi yang terhubung ke Firebase akan memperbarui data secara langsung melalui perangkat yang didaftarkan, baik di situs web atau *smartphone*. Firebase memiliki *library* yang lengkap untuk sebagian besar *platform web* dan *smartphone*, dan dapat dikombinasikan dengan berbagai *framework* lain seperti Node, Java, dan Javascript. Application Programming Interface (API) untuk menyimpan dan menyinkronkan data disimpan sebagai bit di *cloud* dalam bentuk JSON (JavaScript Object Notation) dan disinkronkan secara *real time*. Data tersebutlah yang dapat disinkronkan secara *real time* dengan klien yang terhubung [2].

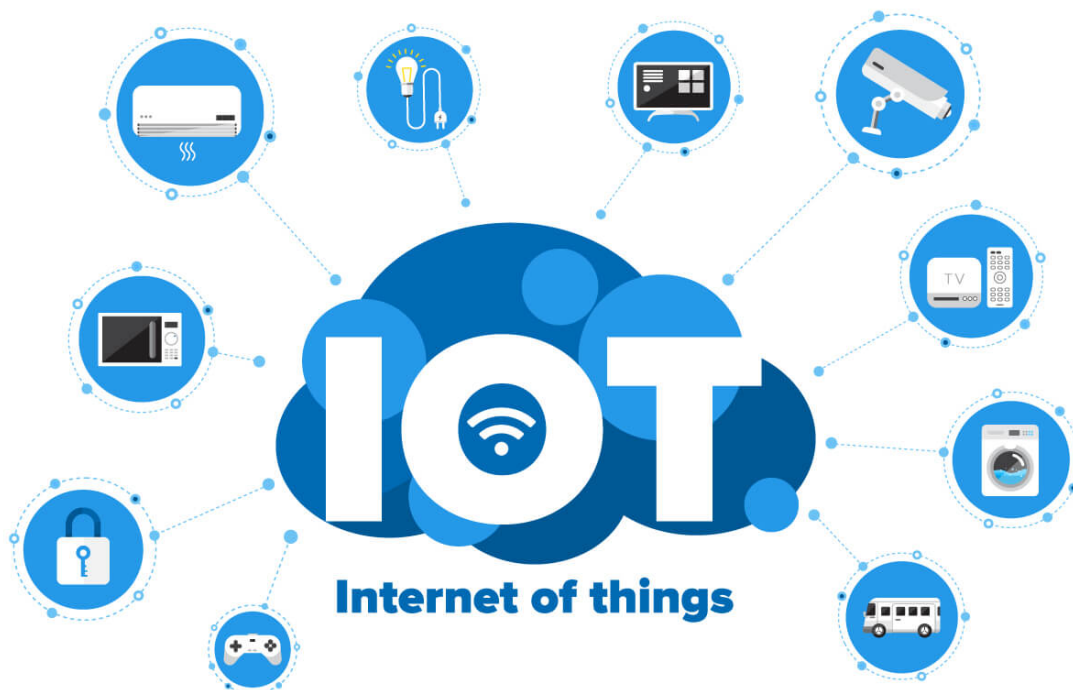


Gambar 2. Firebase (<https://dev.to/>)

Database Firebase adalah database non-relasional atau NoSQL. Basis data ini adalah jenis basis data yang tidak menggunakan sistem tabel dalam implementasinya dan menyimpan data di cloud daripada menyimpannya secara lokal di perangkat [2].

c. Internet of Things

Internet of Things adalah konsep penyematian teknologi seperti sensor dan perangkat lunak ke dalam objek atau objek yang berkomunikasi, mengontrol, menghubungkan, dan bertukar data melalui perangkat lain sambil tetap terhubung ke Internet. IoT erat kaitannya dengan istilah *Machine-to-Machine* atau *M2M*. Semua perangkat dengan kemampuan komunikasi *M2M* sering disebut *smart devices* atau perangkat pintar. Perangkat pintar ini diharapkan dapat membantu manusia dalam menyelesaikan berbagai pekerjaan dan tugas yang ada [3].

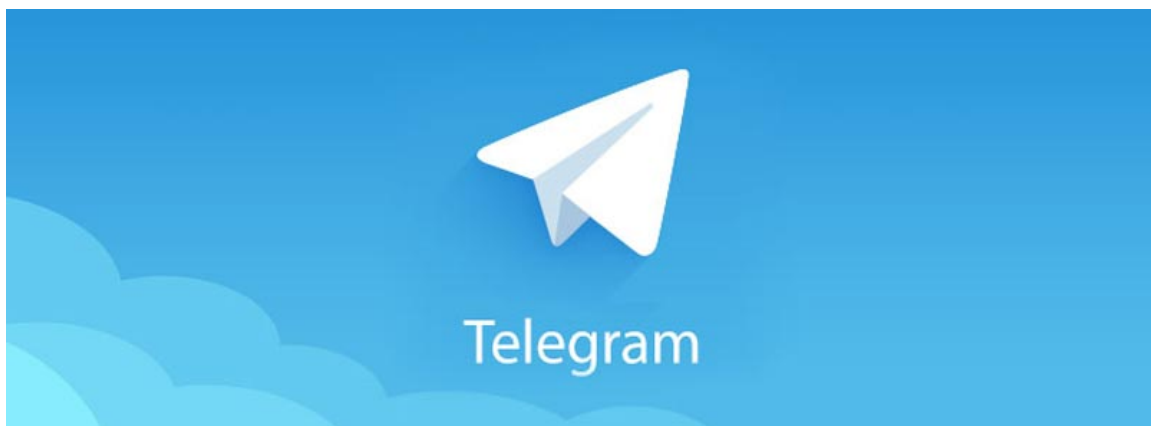


Gambar 3. Internet of Things (<https://www.globalsign.com/>)

Internet of Things adalah konsep penyematian teknologi seperti sensor dan perangkat lunak ke dalam objek atau objek yang berkomunikasi, mengontrol, menghubungkan, dan bertukar data melalui perangkat lain sambil tetap terhubung ke Internet. IoT erat kaitannya dengan istilah *Machine-to-Machine* atau *M2M*. Semua perangkat dengan kemampuan komunikasi *M2M* sering disebut *smart devices* atau perangkat pintar. Perangkat pintar ini diharapkan dapat membantu manusia dalam menyelesaikan berbagai pekerjaan dan tugas yang ada. Untuk membangun IoT, kita tidak hanya membutuhkan perangkat tetapi juga berbagai elemen yang mendukungnya yaitu Sensor, Konektivitas, dan juga *Artificial intelligence* (kecerdasan buatan) [3].

d. Telegram

Telegram adalah aplikasi perpesanan *instan multi-platform* gratis, non-komersial, berbasis *cloud*. Aplikasi Telegram banyak digunakan di berbagai sistem operasi seperti Android, iOS, Windows Phone, Ubuntu Touch dan perangkat komputasi seperti Windows, MacOS X dan Linux. Telegram memungkinkan pengguna untuk saling mengirim pesan teks, foto, video, audio, dokumen, stiker, dan berbagai jenis file lainnya. Telegram dikembangkan oleh Telegram Messenger LLP, didukung oleh pengusaha Rusia Pavel Durov. Kode sisi klien Telegram gratis, tetapi kode sisi servernya bersifat pribadi dan dimiliki oleh perusahaan. Layanan Telegram juga menyediakan API bagi developer (pengembang) untuk membuat stiker animasi, perubahan tampilan, widget, dan bot [4].



Gambar 4. Telegram (<https://www.utopiccomputers.com/>)

Pada Juni 2015, Telegram meluncurkan platform untuk pengembang (pengembang pihak ketiga) untuk membuat akun bot. Bot adalah akun Telegram yang dioperasikan oleh program otomatis. Misalnya, jika pengguna mengirim pesan ke bot menggunakan perintah yang dipahami bot, bot akan langsung merespons pesan tersebut. Bot Telegram dirancang untuk berbagai kebutuhan, mulai dari game, terjemahan bahasa, dan unduhan file hingga pembayaran yang didukung oleh penyedia tertentu seperti Apple Pay, Paymentwall, Yandex.Money, Stripe, dan Ravepay [4].

3. Percobaan

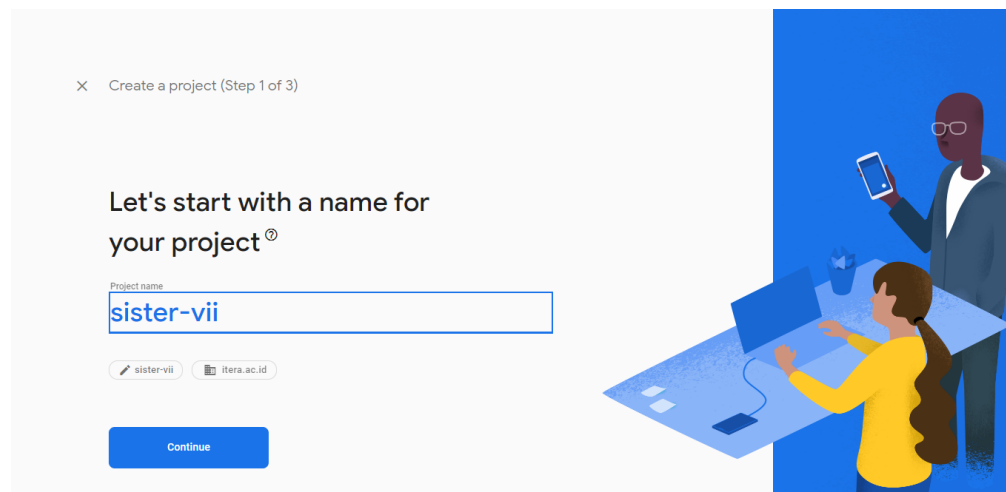
a. Firebase Projects

Pada percobaan ini akan dilakukan konfigurasi Firebase Project untuk dapat menerima dan menyimpan data dari mikrokontroler. Pada percobaan ini akan digunakan sensor DHT11 dan ESP32 sebagai mikrokontrollernya. Sensor tersebut tentu saja dapat diganti sesuai kebutuhan praktikan.

Persiapan

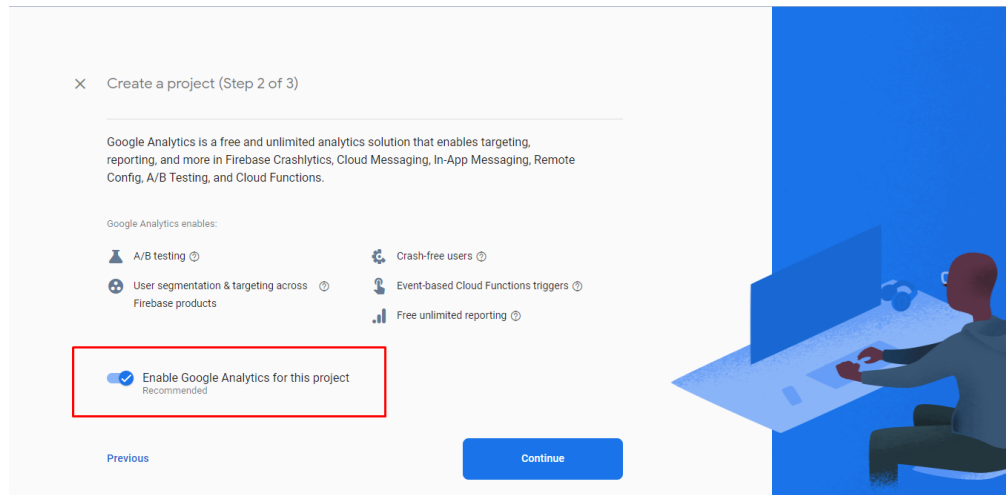
- Konfigurasi Firebase

1. Pertama-tama masuk ke *console* Firebase pada link [Console Firebase](#) kemudian buat project baru dengan nama yang unik contohnya seperti pada **Gambar 1**.

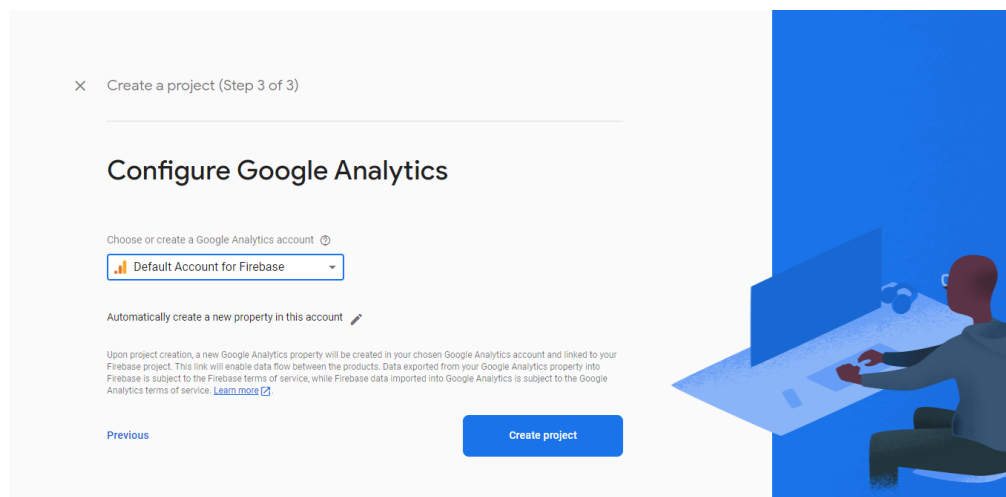


Gambar 1. Membuat Project di Firebase

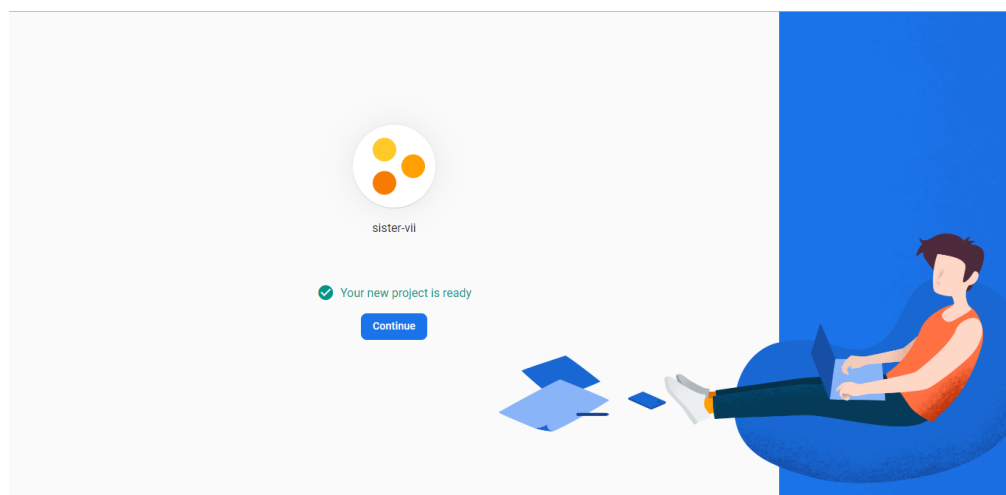
2. Aktifkan **Google Analytic** pada project yang ingin dibuat dan tekan tombol **Continue** untuk ke tahap 3 seperti **Gambar 2**. Pada tahap 3 tinggal pilih user *Google Analytic*nya dan tunggu hingga project selesai dibuat seperti **Gambar 2**.



Gambar 2. Mengaktifkan Google Analytic

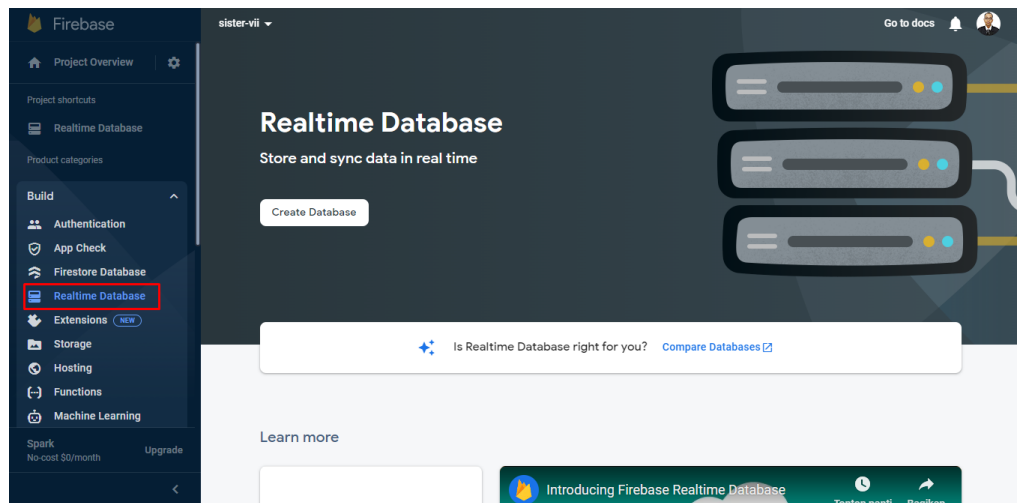


Gambar 3. Pilih Akun



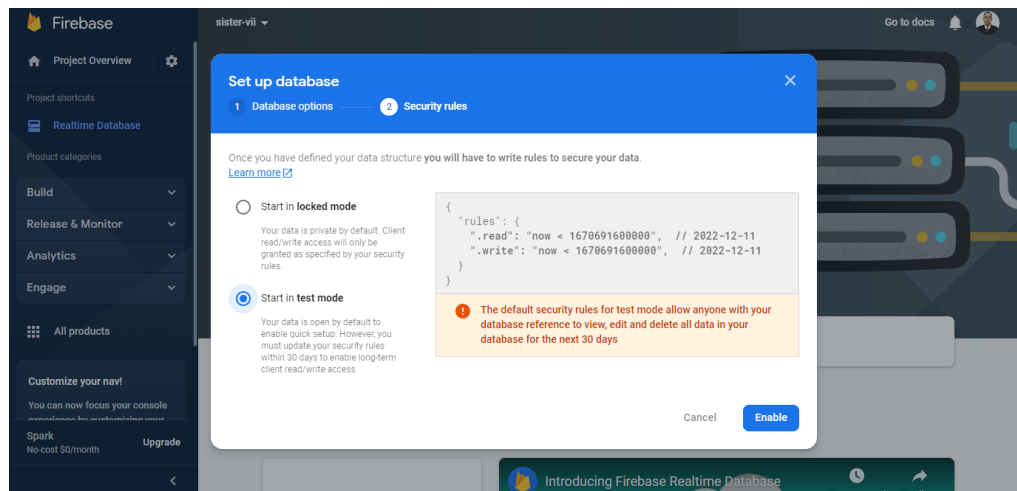
Gambar 4. Project Selesai Dibuat

3. Selanjutnya siapkan terlebih dahulu *database* yang akan kita gunakan melalui *Realtime Database* pada *Build* dan klik *Create Database*.

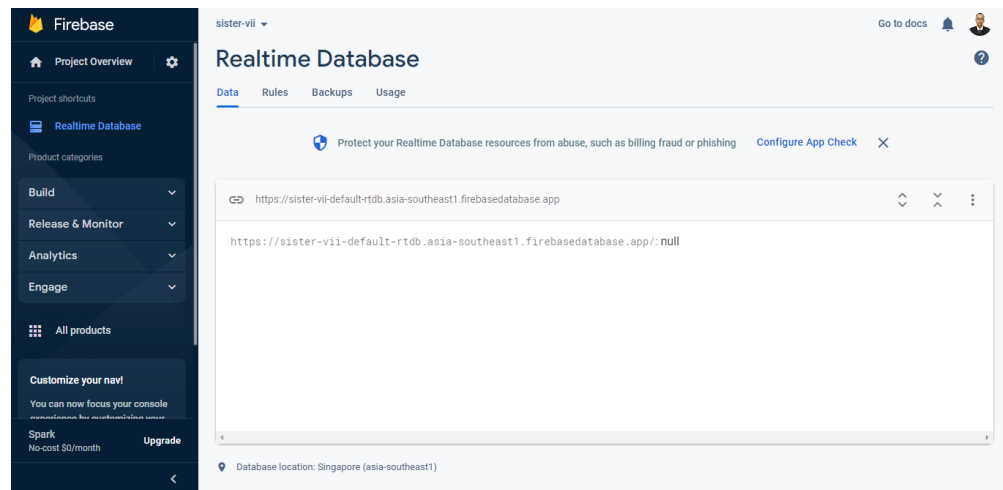


Gambar 5. Membuat Database

4. Atur lokasi database ke US-Central-1 atau asia-southeast1 dan pilih Test Mode, kemudian klik enable untuk membuat *database*.



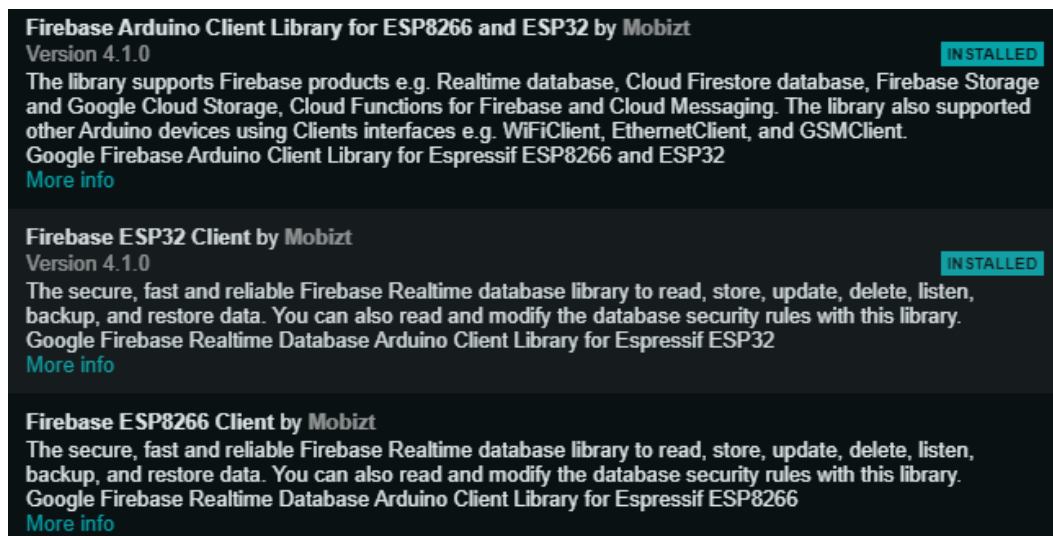
Gambar 6. Test mode Database



Gambar 7. Database berhasil dibuat

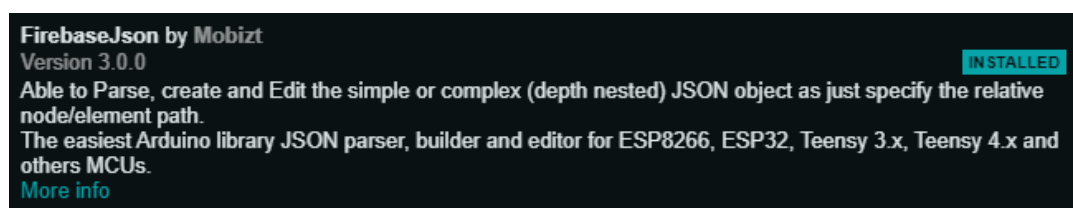
- Install Library
 1. Firebase ESP32 Client

Install terlebih dahulu *library* Firebase ESP32 Client untuk pengguna ESP32, sedangkan pengguna WiFi Module bisa menggunakan Firebase ESP8266 (pada umumnya WiFi module menggunakan ESP8266).



Gambar 8. Library Firebase Client

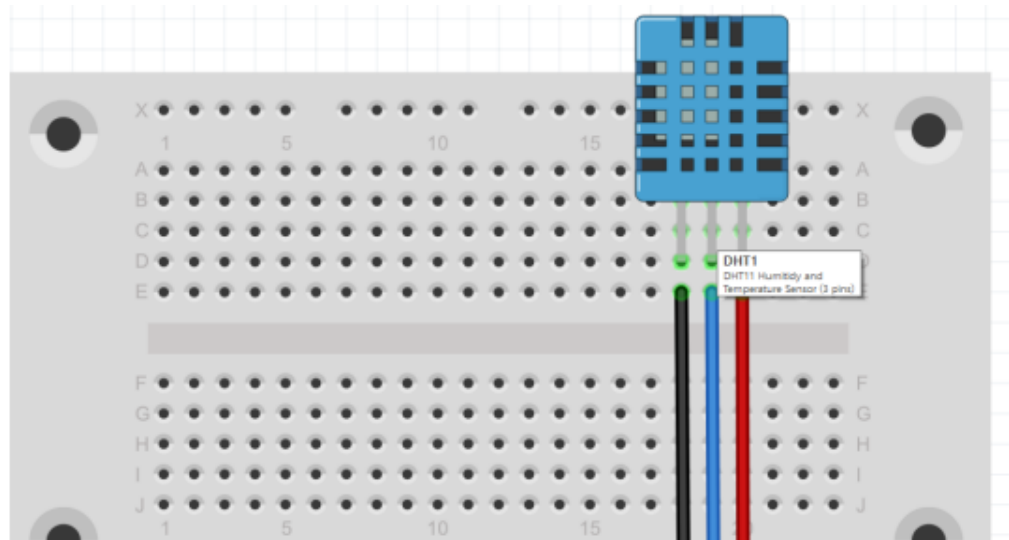
2. FirebaseJson



Gambar 9. Library FirebaseJson

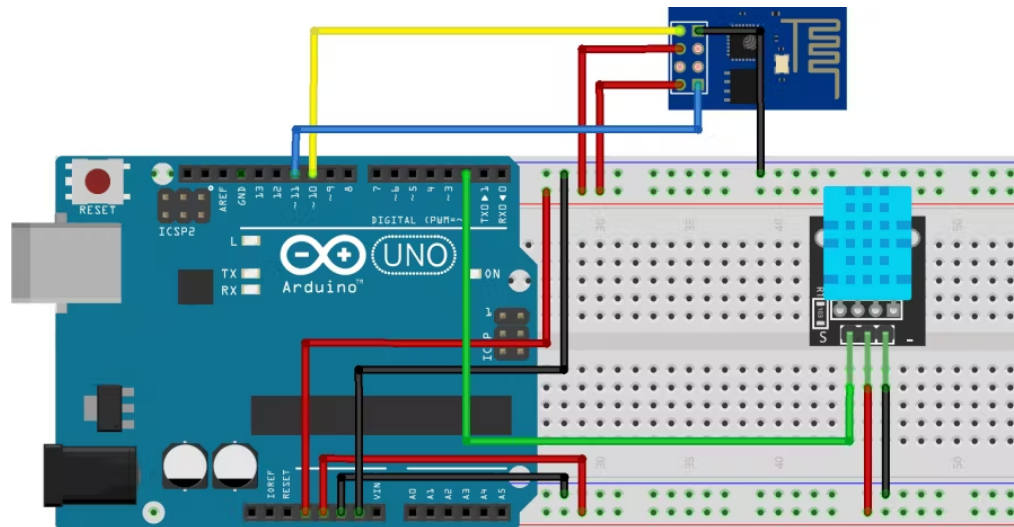
Rangkaian

- ESP32



Gambar 10. Wiring Diagram DHT11

- Arduino Uno + WiFi Module (**Baca BAB 1 Terlebih Dahulu**)



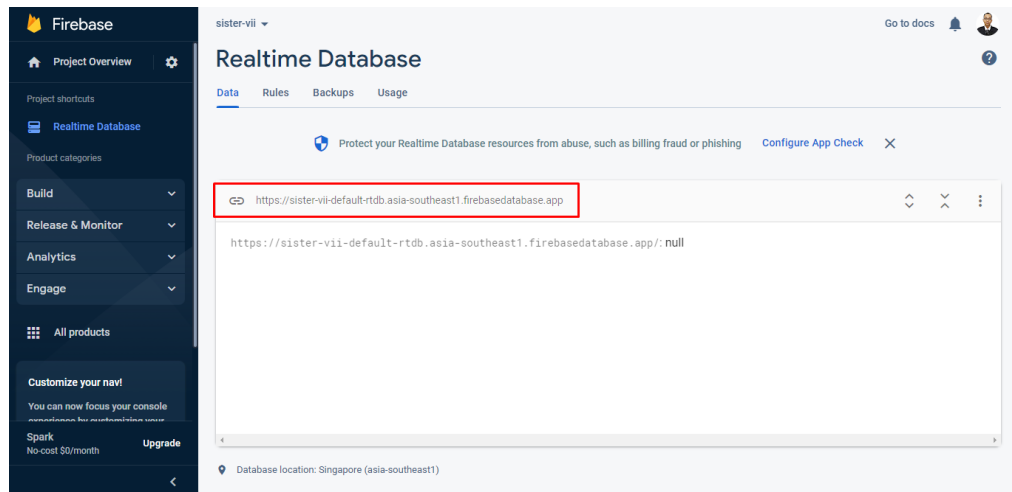
Gambar 11. Wiring Diagram DHT11 + Arduino Uno + WiFi Module ESP01

Code

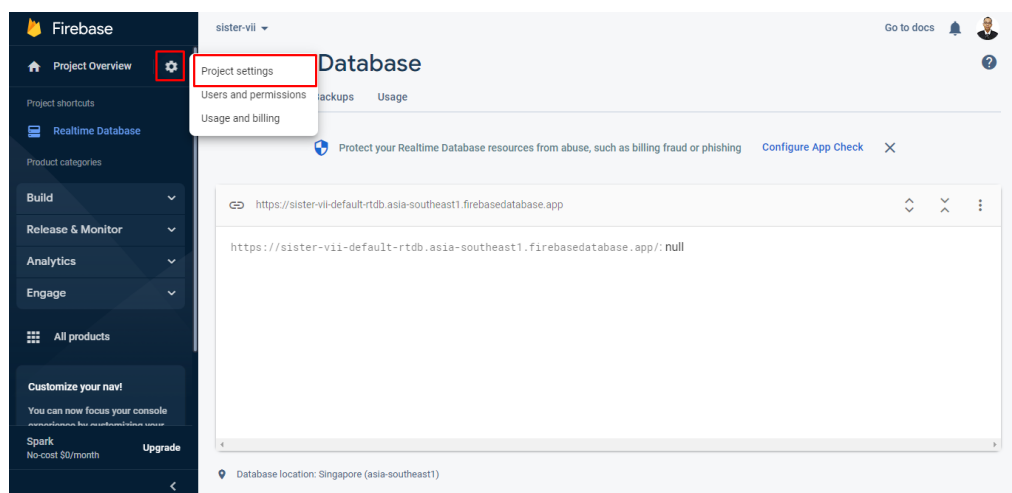
- ReadME First 😊

FIREBASE_HOST, FIREBASE_Authorization_key, WIFI_SSID, dan WIFI_PASSWORD dibutuhkan untuk setiap kode. Oleh sebab itu harus ditentukan terlebih dahulu nilai-nilai tiap variabelnya.

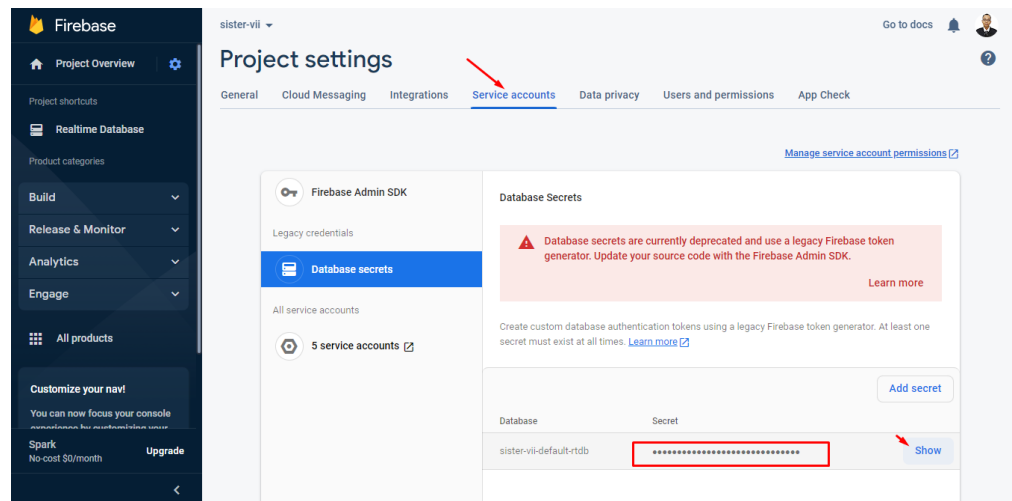
Untuk FIREBASE_HOST dapat diambil dari link realtime database seperti **Gambar 12**, kemudian FIREBASE_Authorization_key diambil dari key pada *project setting* seperti **Gambar 14**, sisanya yaitu SSID dan Password adalah nama dan password WiFi yang tersedia.



Gambar 12. firebase_host



Gambar 13. Project setting



Gambar 14. FIREBASE_Authorization_key

- ESP32
 1. `#include <FirebaseESP32.h>`
 2. `#include <WiFi.h>`
 3. `#include "DHT.h"`
 - 4.
 5. `// Firebase`
 6. `#define FIREBASE_HOST "XXXXXXX"`
 7. `#define FIREBASE_Authorization_key "XXXXXXXXXXXXXXXXX"`
 8. `// Koneksi`
 9. `#define WIFI_SSID "XXXXXX" // Change the name of your WIFI`
 10. `#define WIFI_PASSWORD "XXX" // Change the password of your WIFI`
 - 11.
 12. `#define DHTPIN 14`
 - 13.
 14. `#define DHTTYPE DHT11`
 15. `DHT dht(DHTPIN, DHTTYPE);`
 - 16.
 17. `FirebaseData firebaseData;`
 18. `FirebaseJson json;`
 - 19.
 20. `void setup() {`
 - 21.
 22. `Serial.begin(115200);`
 23. `dht.begin();`
 24. `WiFi.begin(WIFI_SSID, WIFI_PASSWORD);`
 25. `Serial.print("Connecting...");`
 26. `while (WiFi.status() != WL_CONNECTED)`
 27. `{`
 28. `Serial.print(".");`
 29. `delay(300);`
 30. `}`
 31. `Serial.println();`
 32. `Serial.print("IP Address: ");`

```

33. Serial.println(WiFi.localIP());
34. Serial.println();
35. Firebase.begin(FIREBASE_HOST, FIREBASE_Authorization_key);
36.
37.}
38.
39.void loop() {
40.
41. float hum = dht.readHumidity();
42. float temp = dht.readTemperature();
43.
44. if (isnan(hum) || isnan(temp) ){
45.   Serial.println(F("Failed to read from DHT sensor!"));
46.   return;
47. }
48.
49. Serial.print("Temperature: ");
50. Serial.print(temp);
51. Serial.print("°C");
52. Serial.print(" Humidity: ");
53. Serial.print(hum);
54. Serial.print("%");
55. Serial.println();
56.
57. Firebase.setFloat(firebaseData, "/ESP32_APP/TEMPERATURE",
    temp);
58. Firebase.setFloat(firebaseData, "/ESP32_APP/HUMIDITY", hum);
59. delay(200);
60.}

```

- Arduino Uno

```

1. #include <ESP8266WiFi.h>
2. #include <FirebaseArduino.h>
3. #include <DHT.h>
4.
5. #define FIREBASE_HOST "XXXX"
6. #define FIREBASE_AUTH "XXXX"
7. #define WIFI_SSID "XXXX"
8. #define WIFI_PASSWORD "XXXX"
9.
10. #define DHTPIN D2
    // Digital pin connected to DHT11
11. #define DHTTYPE DHT11
    // Initialize dht type as DHT 11
12. DHT dht(DHTPIN, DHTTYPE);
13.
14. void setup()
15. {
16.   Serial.begin(115200);
17.   dht.begin();
    //reads dht sensor data
18.
19.   WiFi.begin(WIFI_SSID, WIFI_PASSWORD);

```

```

20. Serial.print("Connecting to ");
21. Serial.print(WIFI_SSID);
22. while (WiFi.status() != WL_CONNECTED) {
23.   Serial.print(".");
24.   delay(500);
25. }
26.
27. Serial.println();
28. Serial.print("Connected");
29. Serial.print("IP Address: ");
30. Serial.println(WiFi.localIP());
   //prints local IP address
31. Firebase.begin(FIREBASE_HOST, FIREBASE_AUTH);
   // connect to the firebase
32.
33.}
34.
35.void loop()
36.{
37. float h = dht.readHumidity();
   // Read Humidity
38. float t = dht.readTemperature();
   // Read temperature
39.
40. if (isnan(h) || isnan(t))
   // Checking sensor working
41. {
42.   Serial.println(F("Failed to read from DHT sensor!"));
43.   return;
44. }
45. Serial.print("Humidity: ");
46. Serial.print(h);
47. String fireHumid = String(h) + String("%");
   //Humidity integer to string conversion
48.
49. Serial.print("% Temperature: ");
50. Serial.print(t);
51. Serial.println("°C ");
52. String fireTemp = String(t) + String("°C");
   //Temperature integer to string conversion
53. delay(5000);
54.
55.
56. Firebase.pushString("/DHT11/Humidity", fireHumid);
   //setup path to send Humidity readings
57. Firebase.pushString("/DHT11/Temperature", fireTemp);
   //setup path to send Temperature readings
58.   if (Firebase.failed())
59.   {
60.
61.     Serial.print("pushing /logs failed:");
62.     Serial.println(Firebase.error());
63.     return;
64.   }
65.}

```

Pertanyaan

1. Ketika mengirim data ke Firebase, adakah perbedaan antara perintah PUSH (misal `pushFloat`) dengan SET/PUT (misal `setFloat`)?
2. Perintah PUSH dan SET digunakan untuk menyimpan data ke Firebase, perintah apa yang digunakan untuk membaca data dari Firebase?
3. Selain string dan float, tipe data apa saja yang dapat diterima oleh database Firebase?
4. Pada contoh kode sebelumnya, library Firebase bekerja secara synchronous, apa yang perlu diubah agar kode tersebut dapat bekerja secara asynchronous? Lalu, apa kelemahan dari pengiriman data secara asynchronous?
5. Dengan hanya menggunakan library Firebase tersebut (tanpa library/modul tambahan), apakah ada fungsi untuk menyimpan waktu pengiriman data ke database Firebase?
6. Pada contoh kode sebelumnya, database cukup rentan untuk diretas oleh pihak tidak bertanggung jawab karena tidak memiliki autentikasi khusus. Menurutmu, autentikasi apa yang dapat diterapkan dengan library Firebase tersebut?

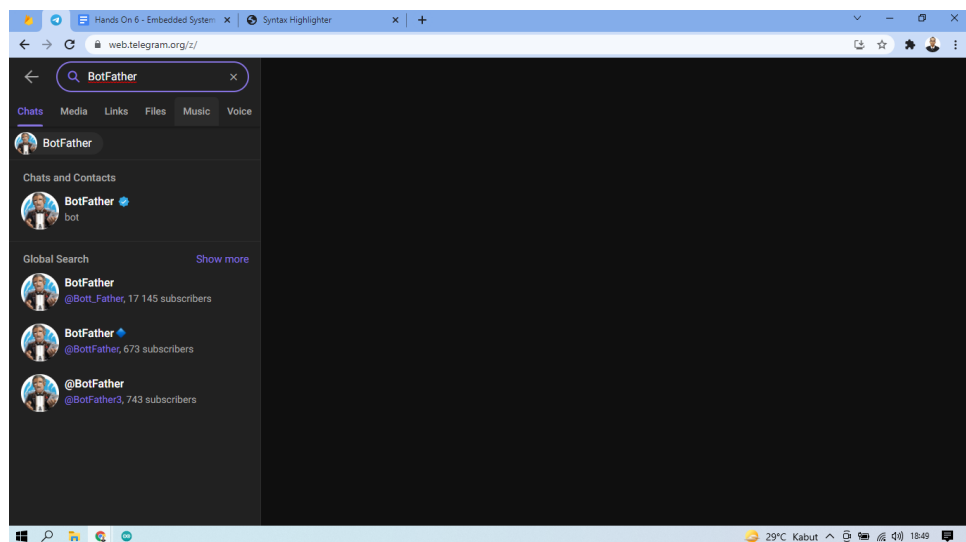
b. Simple BOT

Pada percobaan kali ini dilakukan pembuatan bot pada telegram dan pemrograman dengan Arduino IDE pada papan ESP32 yang mana tujuan dari percobaan kali ini adalah mencoba pengiriman data dari bot telegram ke ESP32.

Persiapan

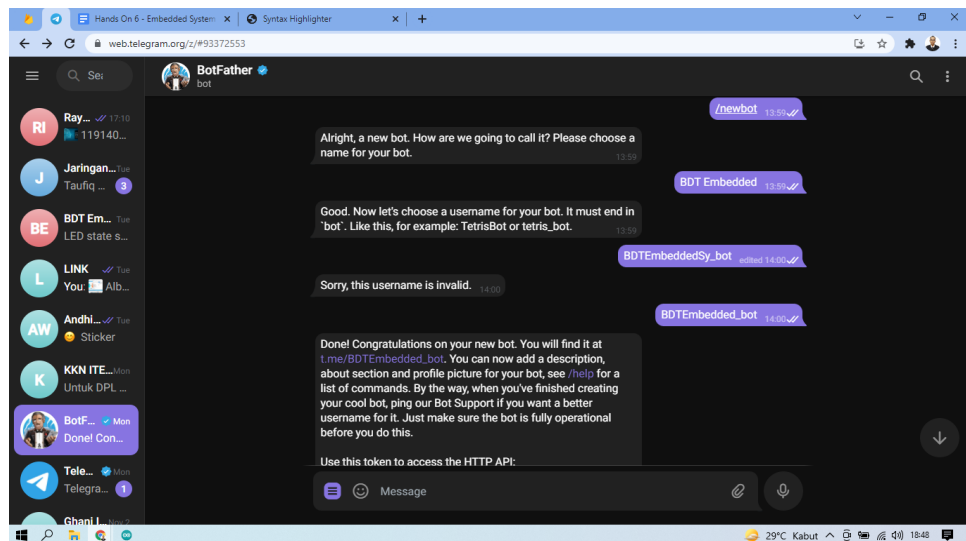
- Pembuatan BOT Telegram

1. Masuk ke dalam aplikasi Telegram, baik itu melalui smartphone ataupun website dan cari BotFather kemudian mulai pembicaraan dengan bot tersebut dengan perintah /start.



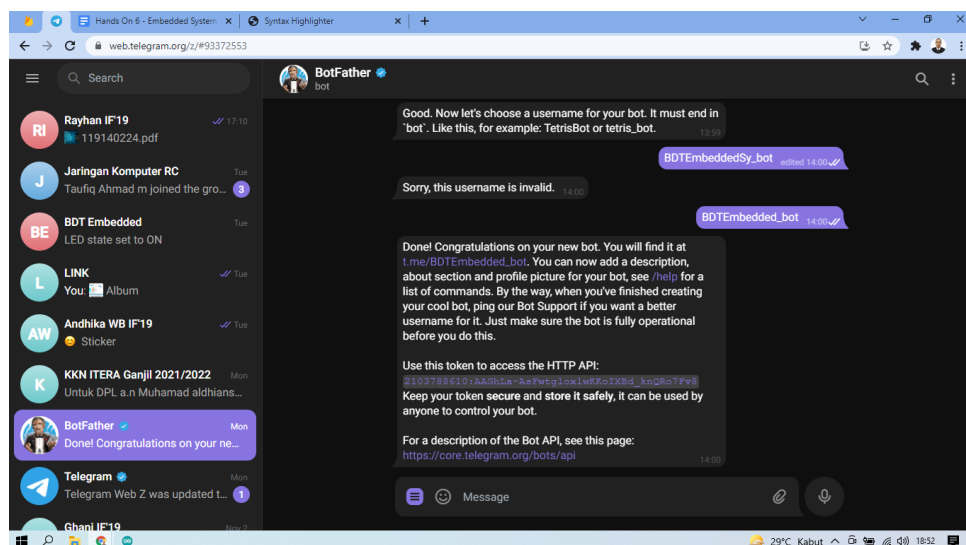
Gambar 15. BotFather Telegram

2. Buat bot sesuai dengan perintah yang tersedia dan jangan lupa untuk menggunakan nama yang unik.



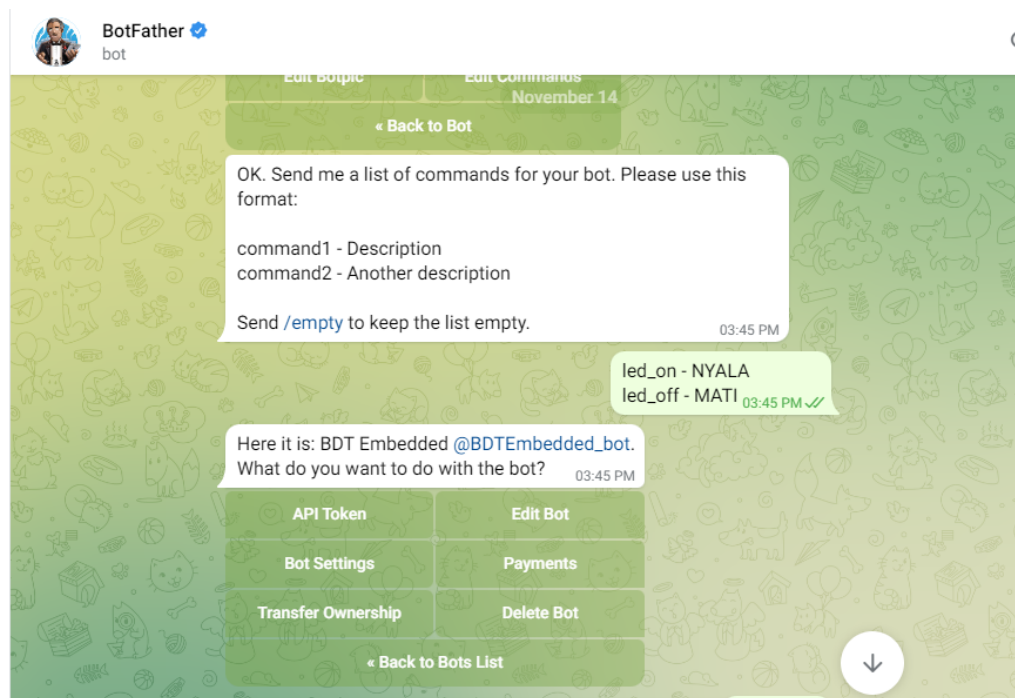
Gambar 16. Rangkaian Sensor DHT11

3. Jika berhasil maka akan diberikan link bot yang telah dibuat dan jangan lupa untuk menyimpan HTTP API bot untuk dimasukkan ke dalam script.



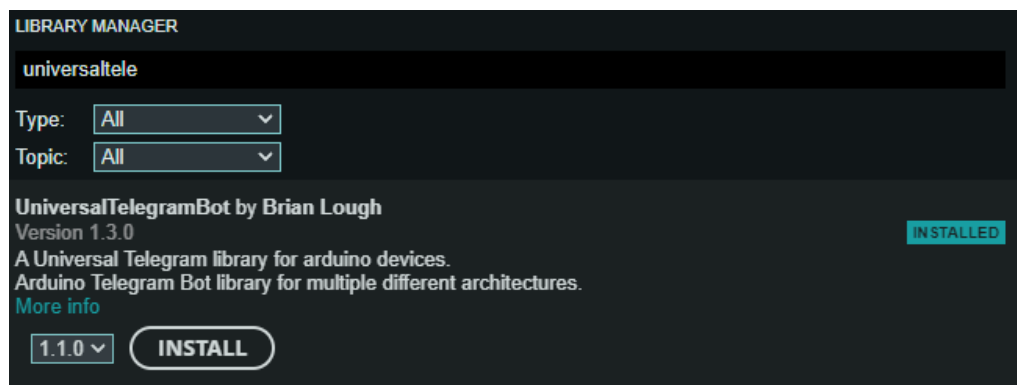
Gambar 17. Bot berhasil dibuat

4. Atur command BOT pada menu edit bot di BotFather

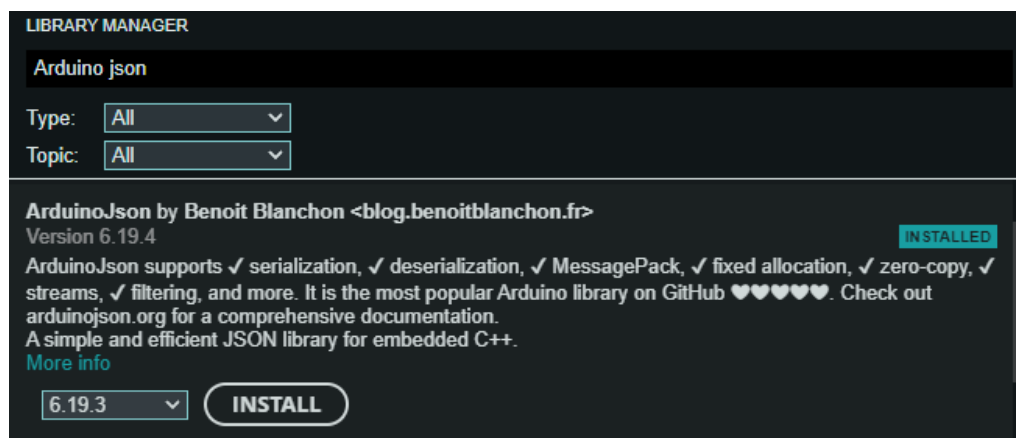


- Install Library

1. UniversalTelegramBot



2. ArduinoJson.



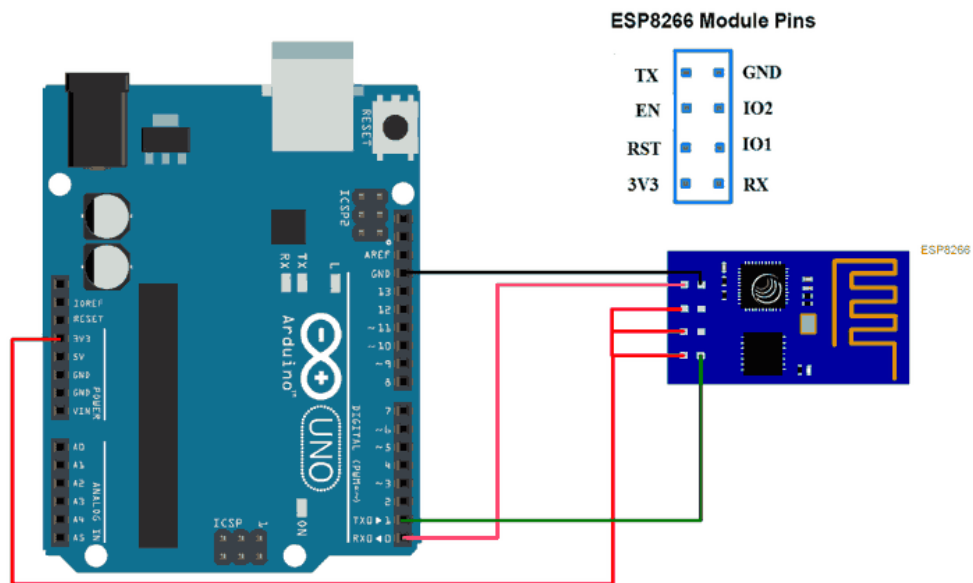
Rangkaian

- ESP32

Pengguna ESP32 dapat langsung menggunakan *On Board* LED (GPIO 2) sehingga tidak perlu ada rangkaian tambahan.

- Arduino Uno

Dapat menggunakan rangkaian percobaan sebelumnya (hilangkan DHT11)



Code

- ESP32 dan Arduino Uno

```

1. #ifndef ESP32
2. #include <WiFi.h>
3. #else
4. #include <ESP8266WiFi.h>
5. #endif
6. #include <WiFiClientSecure.h>
7. #include <UniversalTelegramBot.h> // Universal Telegram Bot
   Library written by Brian Lough:
   https://github.com/witnessmenow/Universal-Arduino-Telegram-Bo
   t
8. #include <ArduinoJson.h>
9.
10. // Replace with your network credentials
11. const char* ssid = "XX";
12. const char* password = "XX";
13.
14. // Initialize Telegram BOT
15. #define BOTtoken "XX" // your Bot Token (Get from Botfather)

```

```

16.
17. #ifdef ESP8266
18. X509List cert(TELEGRAM_CERTIFICATE_ROOT);
19. #endif
20.
21. WiFiClientSecure client;
22. UniversalTelegramBot bot(BOTtoken, client);
23.
24. // Checks for new messages every 1 second.
25. int botRequestDelay = 1000;
26. unsigned long lastTimeBotRan;
27.
28. const int ledPin = 2;
29. bool ledState = LOW;
30.
31. // Handle what happens when you receive new messages
32. void handleNewMessages(int numNewMessages) {
33.   Serial.println("handleNewMessages");
34.   Serial.println(String(numNewMessages));
35.
36.   for (int i=0; i<numNewMessages; i++) {
37.     // Chat id of the requester
38.     String chat_id = String(bot.messages[i].chat_id);
39.
40.     // Print the received message
41.     String text = bot.messages[i].text;
42.     Serial.println(text);
43.
44.     String from_name = bot.messages[i].from_name;
45.
46.     if (text == "/start") {
47.       String welcome = "Welcome, " + from_name + ".\n";
48.       welcome += "Use the following commands to control your
         outputs.\n\n";
49.       welcome += "/led_on to turn GPIO ON \n";
50.       welcome += "/led_off to turn GPIO OFF \n";
51.       welcome += "/state to request current GPIO state \n";
52.       welcome += "/option to return the reply keyboard \n";
53.       bot.sendMessage(chat_id, welcome, "");
54.     }
55.
56.     if (text == "/led_on") {
57.       bot.sendMessage(chat_id, "LED state set to ON", "");
58.       ledState = HIGH;
59.       digitalWrite(ledPin, ledState);
60.     }
61.
62.     if (text == "/led_off") {
63.       bot.sendMessage(chat_id, "LED state set to OFF", "");
64.       ledState = LOW;
65.       digitalWrite(ledPin, ledState);

```

```

66.  }
67.
68.  if (text == "/state") {
69.    if (digitalRead(ledPin)){
70.      bot.sendMessage(chat_id, "LED is ON", "");
71.    }
72.    else{
73.      bot.sendMessage(chat_id, "LED is OFF", "");
74.    }
75.  }
76.
77.  if (text == "/option") {
78.    String keyboardJson = "[[\"/led_on\",
79.      \"/led_off\"],[\"/state\"]]";
80.    bot.sendMessageWithReplyKeyboard(chat_id, "Choose one of
81.      the following options", "", keyboardJson, true);
82.  }
83.
84. void setup() {
85.   Serial.begin(115200);
86.
87.   #ifdef ESP8266
88.     configTime(0, 0, "pool.ntp.org"); // get UTC time via
89.     NTP
90.     client.setTrustAnchors(&cert); // Add root certificate for
91.     api.telegram.org
92.   #endif
93.   pinMode(ledPin, OUTPUT);
94.   digitalWrite(ledPin, ledState);
95.   // Connect to Wi-Fi
96.   WiFi.mode(WIFI_STA);
97.   WiFi.begin(ssid, password);
98.   #ifdef ESP32
99.     client.setCACert(TELEGRAM_CERTIFICATE_ROOT); // Add root
100.    certificate for api.telegram.org
101.   #endif
102.   while (WiFi.status() != WL_CONNECTED) {
103.     delay(1000);
104.     Serial.println("Connecting to WiFi..");
105.   }
106.   // Print ESP32 Local IP Address
107.   Serial.println(WiFi.localIP());
108. }
109. void loop() {
110.   if (millis() > lastTimeBotRan + botRequestDelay) {

```

```
111.     int numNewMessages =
        bot.getUpdates(bot.last_message_received + 1);
112.
113.     while(numNewMessages) {
114.         Serial.println("got response");
115.         handleNewMessages(numNewMessages);
116.         numNewMessages =
            bot.getUpdates(bot.last_message_received + 1);
117.     }
118.     lastTimeBotRan = millis();
119. }
120. }
```

Pertanyaan

1. Mengapa fungsi `setTrustAnchors` dan `setCACert` diperlukan untuk berkomunikasi dengan Telegram? Apa guna certificate pada fungsi tersebut?
2. Apa perbedaan fungsi `sendMessage`, `sendMessageWithReplyKeyboard`, dan `sendMessageWithInlineKeyboard`?
3. Pada fungsi `sendMessage` pada kode di atas, pesan dikirim dalam bentuk plain teks biasa, adakah cara untuk mengirim teks dengan format tertentu (misal markdown atau HTML)?
4. Perintah (fungsi) apa yang dapat digunakan untuk mengirim foto dari bot Telegram?

4. Challenge

a. Simple Read Write (Firebase)

Buatlah sebuah pengaplikasian kasus secara bebas dengan menggunakan fungsi penyimpanan dan pembacaan data dari Firebase. Adapun ketentuan yang perlu diterapkan yaitu:

1. Menggunakan setidaknya 1 buah sensor
2. Menggunakan setidaknya 1 buah aktuator
3. Tidak menggunakan lagi DHT (tanpa pengaplikasian sensor/aktuator tambahan)
4. Terdapat sebuah **key tambahan** pada database Firebase yang menyimpan value **berupa waktu** (timestamp) pengiriman data
5. Bonus poin tambahan apabila menggunakan metode autentikasi tertentu dan dijelaskan pada langkah pengerjaan/analisis laporan
6. Boleh menyertakan video (misal melalui link) bila malas menggunakan foto. Namun, tidak ada poin tambahan lagi bila video ditambahkan opening, musik, penutup, dsb

b. Subscriber Mode (Telegram)

Buatlah sebuah pengaplikasian kasus secara bebas dengan menggunakan bot Telegram. Adapun ketentuan yang perlu diterapkan yaitu:

1. Menggunakan setidaknya 1 buah sensor
2. Menggunakan setidaknya 1 buah aktuator
3. Tidak menggunakan lagi LED (tanpa pengaplikasian sensor/aktuator tambahan)
4. Terdapat setidaknya 2 buah perintah selain fungsi stub seperti /start, /help, /status, atau /options
5. Terdapat perintah tambahan yaitu **/subscribe** dan **/unsubscribe**, dimana status dari sensor/aktuator akan dikirim tiap 10 detik secara otomatis tanpa perlu menunggu chat dari user (bila user telah subscribe ke bot). Tunggu selama 10 detik tersebut bersifat non-blocking (asynchronous) sehingga bot tetap dapat menerima/melakukan aksi lainnya
6. Bukti /subscribe dan /unsubscribe terlihat secara jelas pada laporan
7. Boleh menyertakan video (misal melalui link) bila malas menggunakan foto. Namun, tidak ada poin tambahan lagi bila video ditambahkan opening, musik, penutup, dsb

Referensi

- [1] P. Peniarsih, "CLOUD COMPUTING TECHNOLOGIES AND BUSINESS OPPORTUNITIES," *JURNAL TEKNOLOGI INDUSTRI*, vol. 1, no. 0, Apr. 2021, doi: 10.35968/jti.v1i0.686.
- [2] E. A. W. Sanad, "Pemanfaatan Realtime Database di Platform Firebase Pada Aplikasi E-Tourism Kabupaten Nabire," *Jurnal Penelitian Enjiniring*, vol. 22, no. 1, pp. 20–26, May 2019, doi: 10.25042/jpe.052018.04.
- [3] R. Setiawan, "Memahami Apa Itu Internet of Things," *Dicoding Blog*, Sep. 07, 2021. <https://www.dicoding.com/blog/apa-itu-internet-of-things/>
- [4] R. D. Arifin, "Apa itu Telegram? Pengertian Telegram Adalah, Sejarah, Fitur, Kelebihan," *Dianisa.com*, May 25, 2022. <https://dianisa.com/pengertian-telegram/> (accessed Nov. 11, 2022).