**Telkom - ITDRI**

**30 September 2022**



Modul Pelatihan

“Peningkatan Kompetensi Mahasiswa Pada Pemanfaatan Platform Internet of Things ”

# DAFTAR ISI

1. [Tujuan Pelatihan 2](#_bookmark0)
2. [Pengenalan Perangkat 2](#_bookmark3)
   1. [Perangkat Keras 2](#_bookmark4)
   2. [Perangkat Lunak 2](#_bookmark5)

[3 Arduino IDE 2](#_bookmark6)

1. [Definisi Arduino IDE 2](#_bookmark7)
2. Instalasi Arduino IDE 3
3. Menambahkan Board ESP32 5
4. Include Library 8
5. Upload Program Example 10
6. [Integrasi Sensor DHT 12](#_bookmark12)
   1. [Spesifikasi DHT 12](#_bookmark13)
   2. [Wiring Diagram 13](#_bookmark14)
   3. [Upload sketch program 14](#_bookmark16)
   4. [Keterangan Program 15](#_bookmark17)
7. [Telkom IoT Platform 16](#_bookmark18)
   1. [Registrasi Akun 16](#_bookmark20)
   2. [Login Akun 17](#_bookmark22)
   3. Create Application 18
   4. Create Device 19
   5. Create Access Key 21
8. [Integrasi Device ke Telkom IoT Platform melalui Protokol LoRa 22](#_bookmark24)
   1. [Arsitektur LoRaWan 22](#_bookmark25)
   2. Upload Sketch Program via USB 22
   3. [Monitoring Data Real Time di Dashboard IoT Platform 25](#_bookmark26)
   4. Keterangan Program 26

**7 Referensi** 28

## Tujuan Pelatihan

## Peserta pelatihan diharapkan mampu mengetahui fungsi dan fitur dari perangkat IoT yaitu ESP32, DHT11 serta mampu mengaplikasikan dalam usecase tertentu.

## Peserta pelatihan diharapkan mampu mengetahui fungsi dan fitur dari Perangkat Lunak Arduino IDE serta mampu mengapikasikan dalam usecase tertentu.

## Peserta pelatihan diharapkan mampu mengeksplorasi fitur-fitur yang ada dalam Dashboard IoT Platform (Registrasi, Create Device, Create Application dan Access Key)

## Peserta pelatihan diharapkan mampu memahami cara mengkoneksikan Perangkat ke IoT Platform menggunakan konektivitas jaringan LoRA.

## Pengenalan Perangkat

## Perangkat Keras

## Developer Shield ESP32 : 1 (satu) buah

## Board Shield Sensor : 1 (satu) buah

## Kabel USB tipe C ke B : 1 (satu) buah

## Laptop : 1 (satu) buah

## Perangkat Lunak

## Arduino IDE

## Library DHT\_sensor\_Library

## Library LoRaWAN-Arduino-Library

## Arduino IDE

## Definisi Arduino IDE

Arduino IDE adalah software yang digunakan untuk membuat sketch pemrogaman atau dengan kata lain arduino IDE sebagai media untuk pemrogaman pada board yang ingin diprogram. Arduino IDE ini berguna untuk mengedit, membuat, meng-upload ke board yang ditentukan, dan meng-coding program tertentu. Arduino IDE dibuat dari bahasa pemrogaman JAVA, yang dilengkapi dengan library C/C++(wiring), yang membuat operasi input/output lebih mudah.

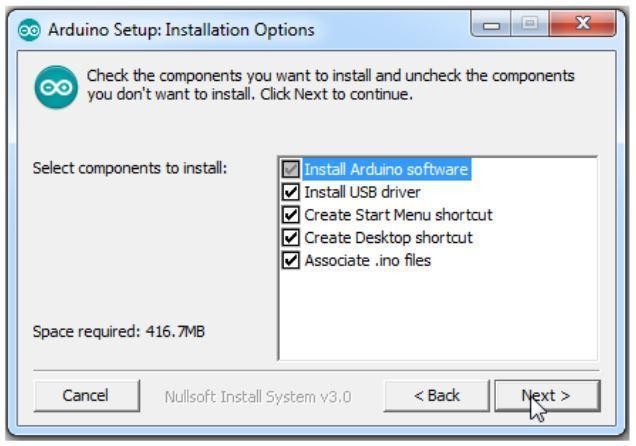
Sketch adalah program yang ditulis dengan menggunakan Arduino IDE. Sketch yang disimpan akan memiliki ekstensi file .ino. Kemudian dalam penulisan program pada arduino IDE ini ada beberapa stuktur dasar.

## Instalasi Arduino IDE

1. Arduino IDE versi terbaru dapat didownload website resmi Arduino ([*https://www.arduino.cc/en/software*](https://www.arduino.cc/en/software))

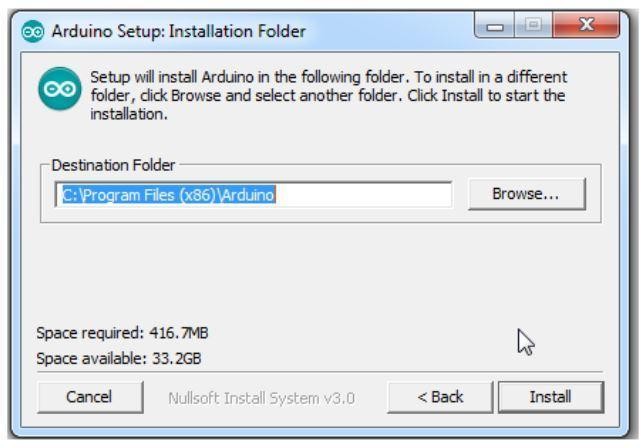


Gambar Opsi Download Arduino IDE

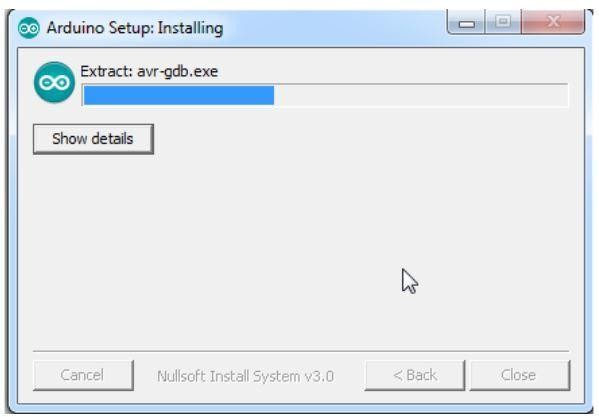
1. Setelah selesai download silahkan buka dengan klik kiri dua kali atau klik kanan open kemudian akan muncul ***License Agreement*** atau Persetujuan Instalasi, klik tombol ***I Agree*** untuk memulai install software Arduino IDE.
2. Untuk ***Installation Option*** pilih semua option dan klik tombol ***Next.***

Gambar. Pilihan Opsi Instalasi.

1. ***Installation Folder atau Pilihan Folder*** untuk memilih folder tempat menyimpan program arduino dan klik tombol **install** untuk memulai proses instalasi software.

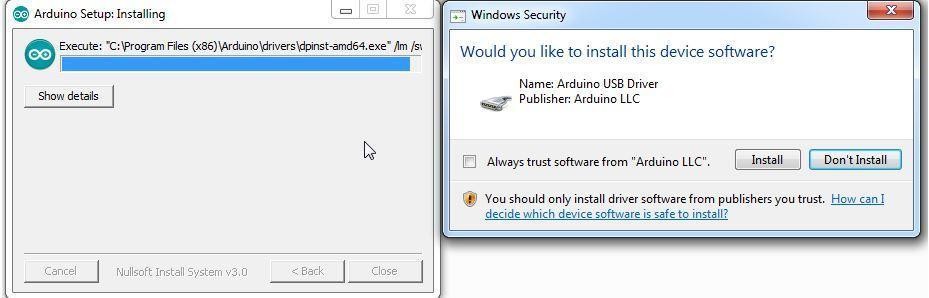


Gambar. Pilihan *Installation Folder* atau Pilihan Folder.

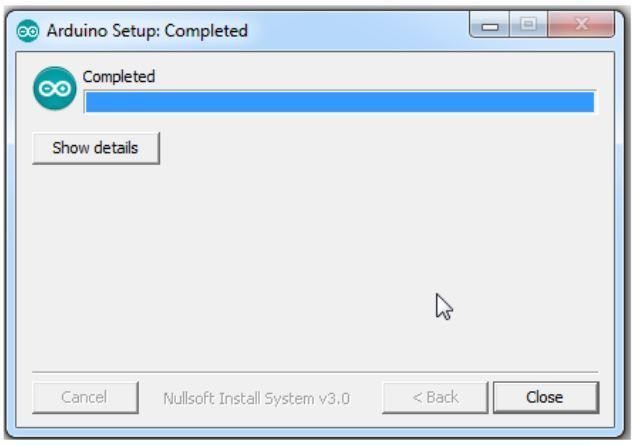
1. Proses Extract dan Instalasi di mulai

Gambar. Proses Extract dan Instalasi di mulai.

1. Saat proses instalasi sedang berlangsung akan muncul pilihan untuk install driver, pilih tombol instal,proses ini untuk mengenali dan melakukan komunikasi dengan board arduino melalui port USB.



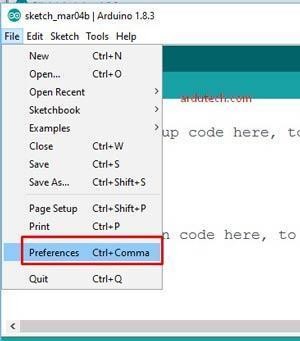
Gambar. Install USB Drive untuk Arduino.

1. Setelah selesai silahkan klik close.

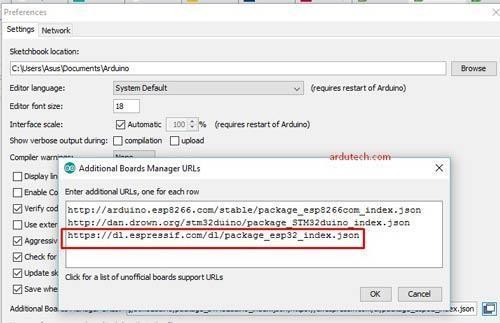
Gambar. Proses Instalasi Selesai.

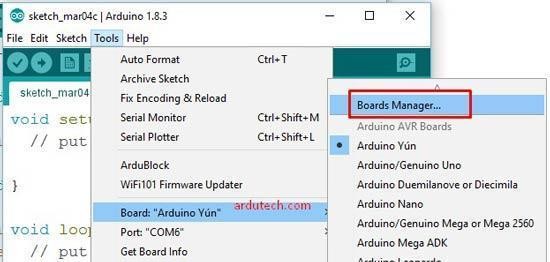
* 1. **Menambahkan Board ESP32**

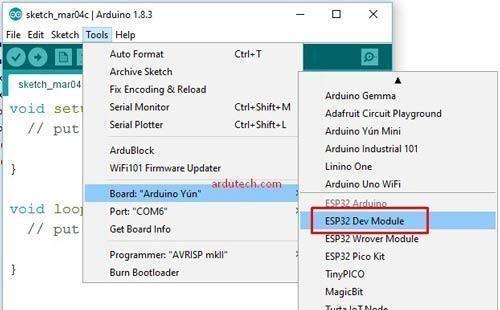
1. Pastikan komputer/laptop terhubung dengan internet. Buka/jalankan Arduino IDE. Dari menu **File > Preferences** akan tampil jendela **Preferences**.



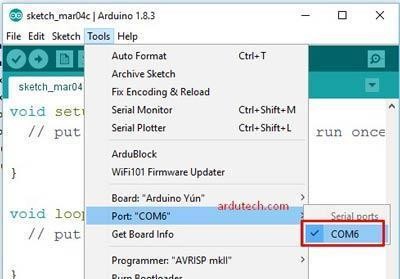
1. Pada kolom “Additional Boards Manager URLs:” isikan :

https://dl.espressif.com/dl/package\_esp32\_index.json

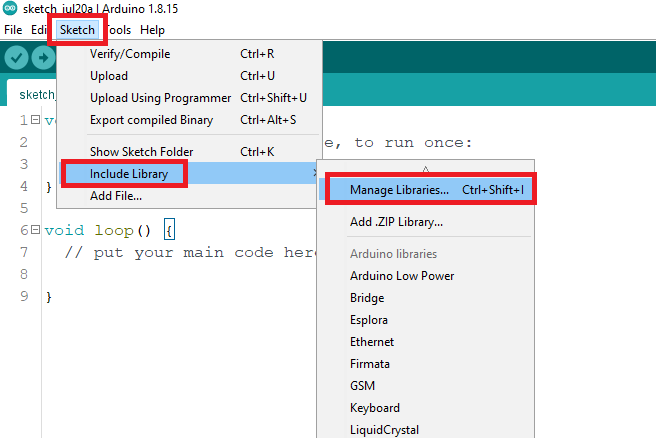
1. Jika sebelumnya sudah ada link lain seperti esp8266 klik kotak disebelah kanan kemudian letakkan link yang baru tadi (esp32) dibagian bawahnya atau dapat juga dengan tanda koma. Kemudian klik “**OK**”.
2. Buka Boards Manager dari menu **Tools > Board > Boards Manager …**
3. Cari **ESP32** pada kolom ***search*** kemudian klik “**Install**”.
4. Tunggu sampai proses *downloading* selesai.
5. Jika *downloading* sudah selesai, hubungkan ESP32 DevKit ke port USB komputer. Pilih jenis boardnya dari menu **Tools > Board > ESP32 Dev Module**.



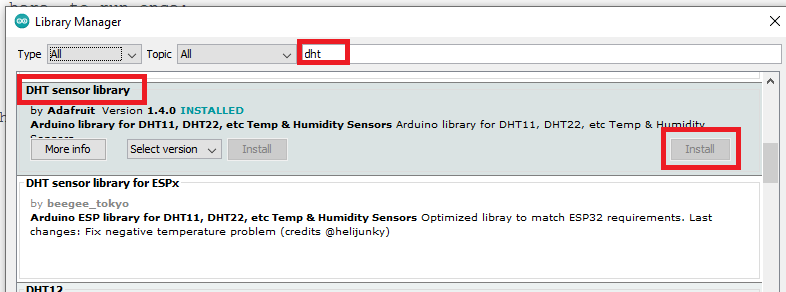
1. Pilih nomor Port-nya, cek di *Device Manager*. Jika port belum terdeteksi, instal terlebih dahulu driver USB (ESP32 DevKit memakai driver USB CH340) Sesuaikan nomor port, dari menu **Tools**

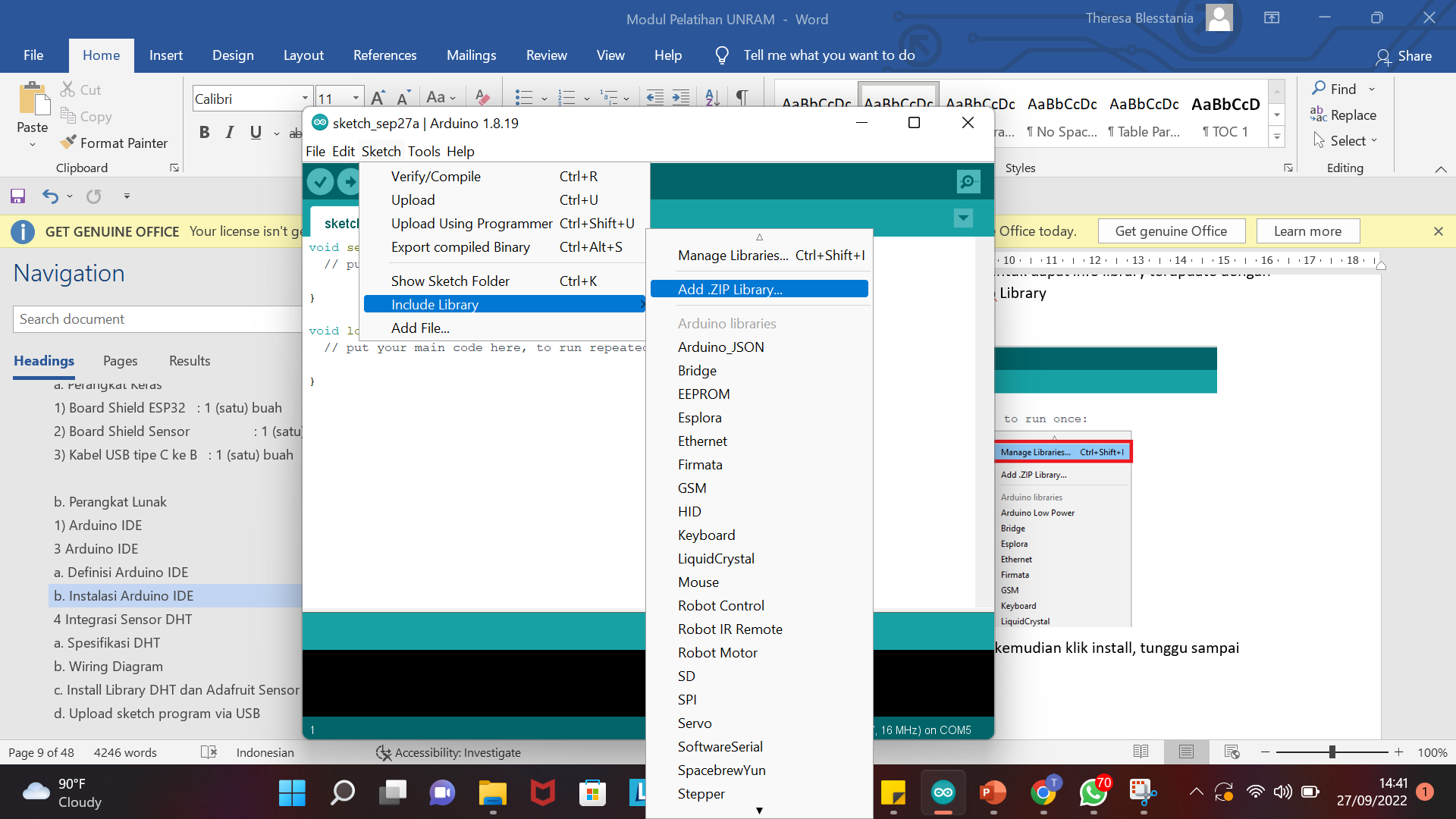
**> Port**

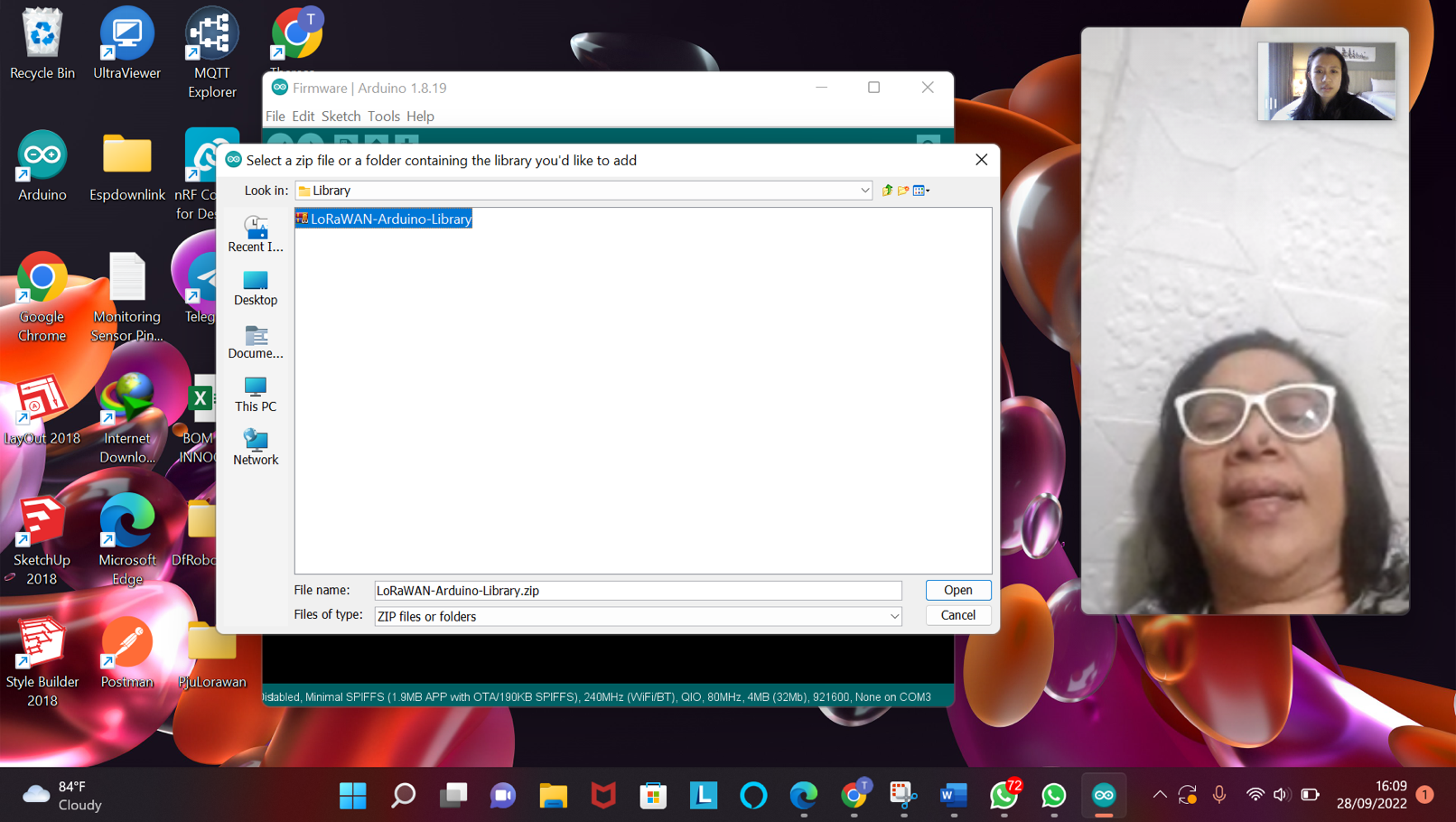
* 1. **Include Library**
     1. Ada 2 library yang dibutuhkan untuk mengakses sensor DHT, yaitu DHT.h dan Library Untuk LoRa yaitu lorawan.h.
     2. Lakukan instalasi menggunakan Manage Libraries untuk dapat info library terupdate dengan cara membuka tab Sketch>include Library>Manage Libraries



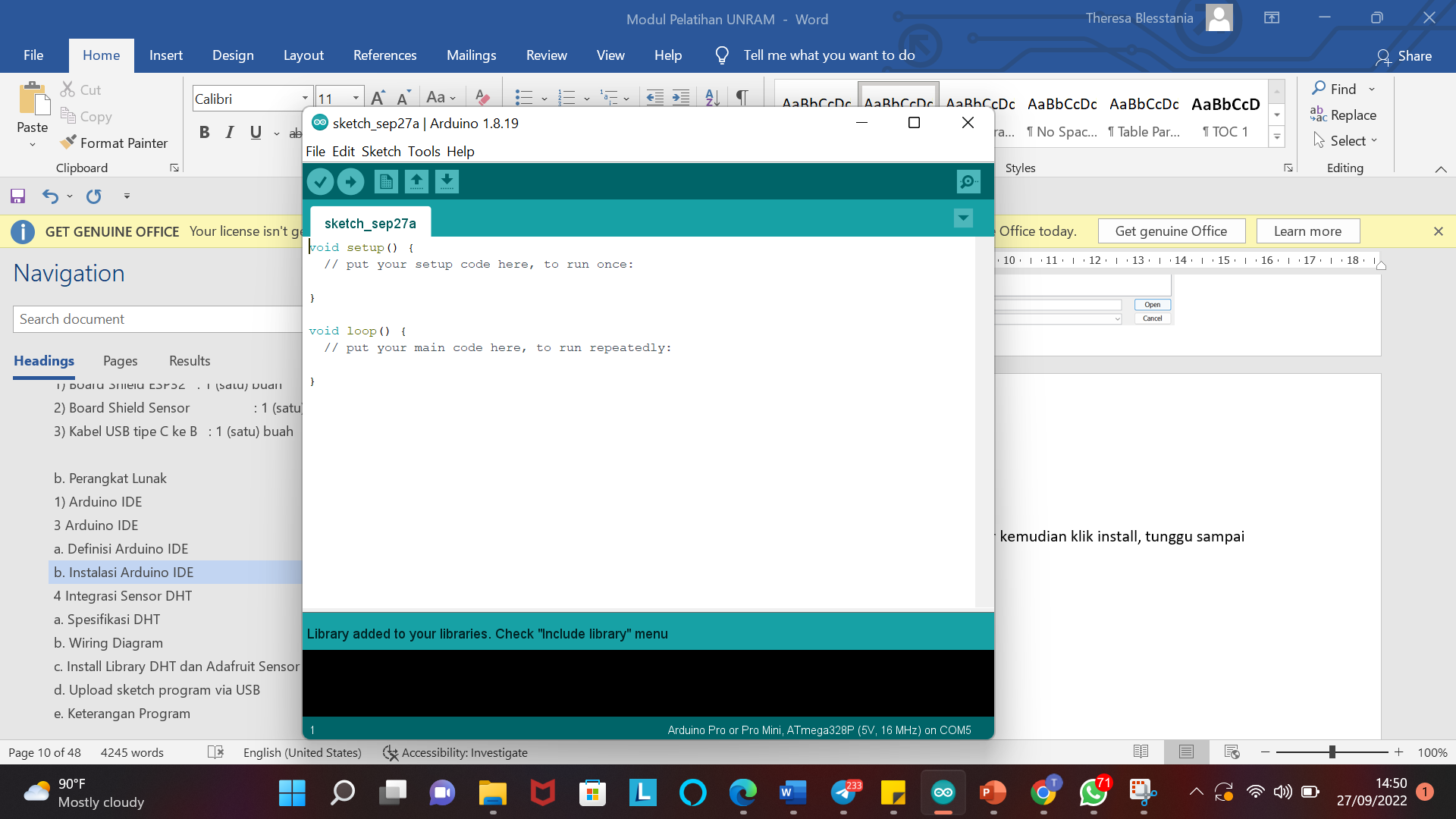
* + 1. Kemudian ketik “dht” lalu pilih DHT Sensor Library kemudian klik install, tunggu sampai

proses instalasi selesa

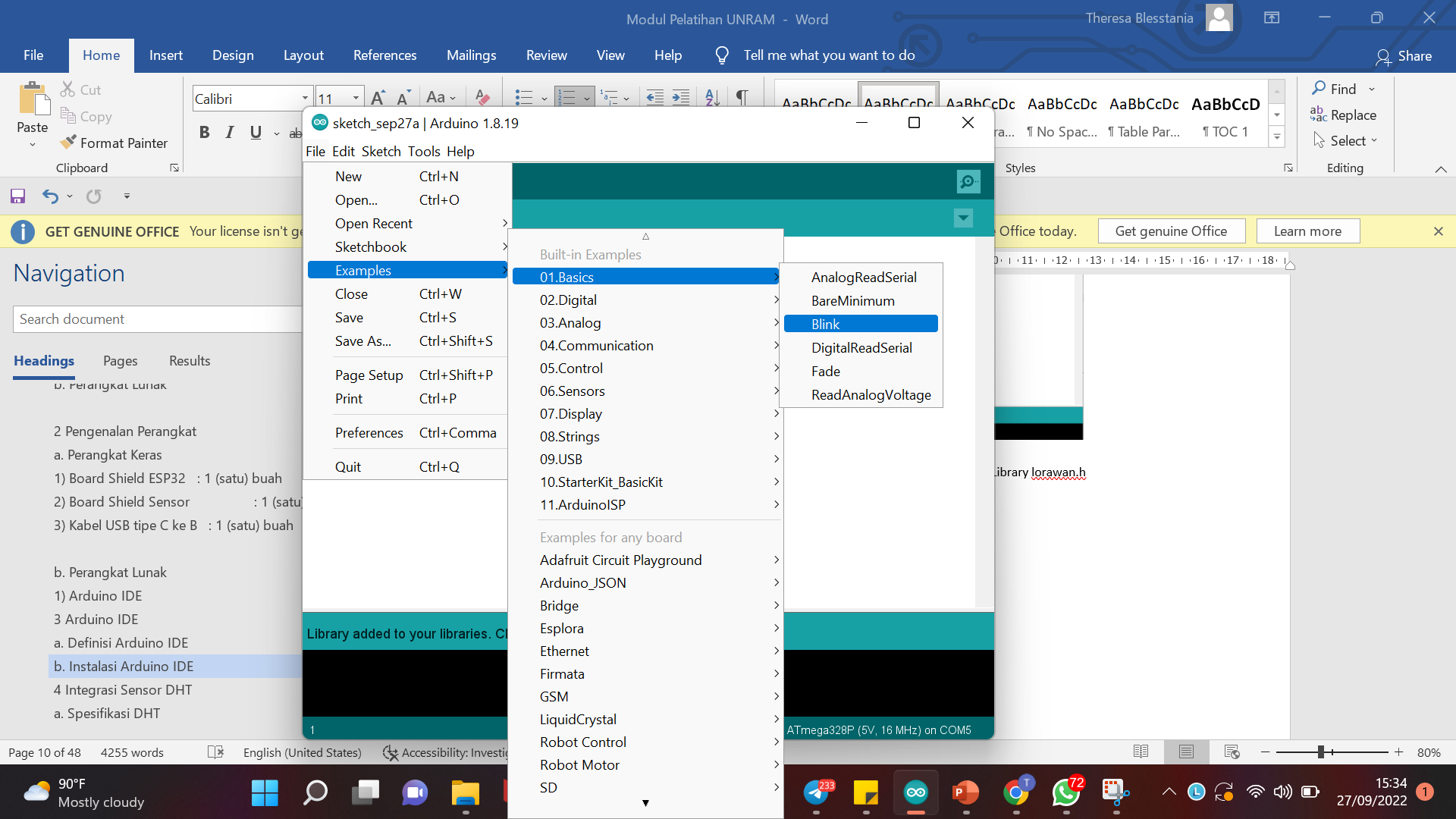
* + 1. Untuk menambahkan Library Lorawan.h dilakukan dengan cara membuka tab Sketch>include Library>Add.Zip Library
    2. Kemudian ketik cari File Library yang tersimpan di dalam File Workshop Unram>Library>LoRaWAN lalu klik Open



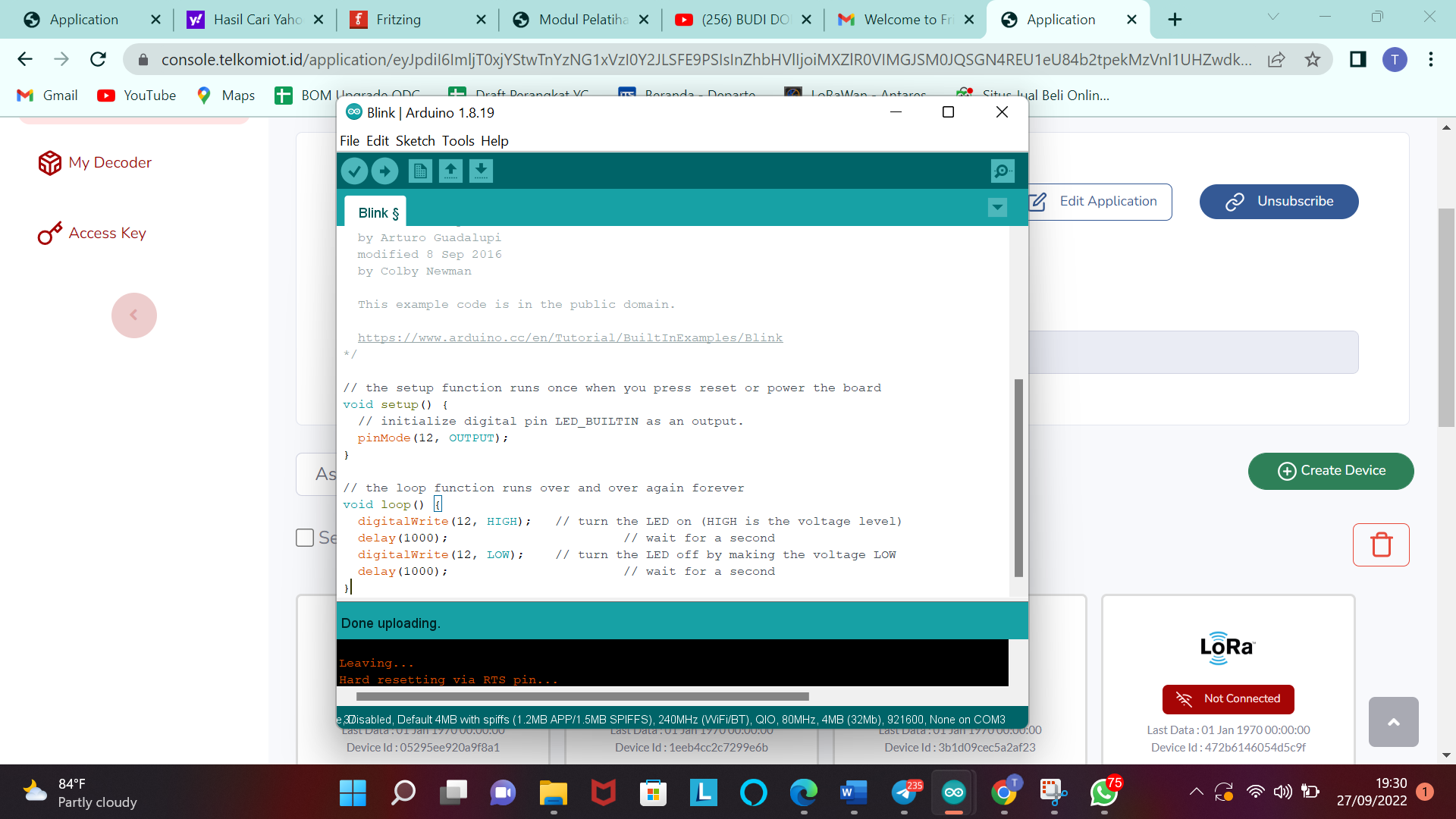
* + 1. Maka Library lorawan.h sudah ditambahkan ke dalam Libraries Arduino IDE



* 1. **Upload Program Example**
     1. Klik Menu File>Examples>01 Basics>Blink



* + 1. Kemudian Program Blink akan ditampilkan pada Sketch



* + 1. Simpan (**Save**) kemudian **Upload**. Tunggu proses *compiling* selesai dan sukses (tidak ada

erorr). Sambungkan kabel data USB dari ESP32 dan Komputer/Laptop, kemudian **Upload** lalu tunggu proses *uploading* selesai dan sukses (mengisikan program dari komputer ke memori ESP32).

* + 1. Jika sudah terupload maka akan terlihat LED pada Board akan Berkedip dengan delay 1 detik.

## Integrasi Sensor DHT

## Spesifikasi DHT

DHT11 adalah salah satu sensor yang dapat mengukur dua parameter lingkungan sekaligus, yakni suhu dan kelembaban udara (humidity). Dalam sensor ini terdapat sebuah thermistor tipe NTC (Negative Temperature Coefficient) untuk mengukur suhu, sebuah sensor kelembaban tipe resisitif dan sebuah mikrokontroller 8-bit yang mengolah kedua sensor tersebut dan mengirim hasilnya ke pin output dengan format single-wire bi-directional (kabel tunggal dua arah). Jadi walaupun kelihatannya kecil, DHT11 ini ternyata melakukan fungsi yang cukup kompleks.

Spesifikasi Pengukuran Kelembaban Udara

* Resolusi pengukuran: 16Bit
* Repeatability: ±1% RH
* Akurasi pengukuran: 25℃ ±5% RH
* Interchangeability: fully interchangeable
* Waktu respon: 1 / e (63%) of 25℃ 6 detik
* Histeresis: <± 0.3% RH
* Long-term stability: <± 0.5% RH / yr in

Spesifikasi Pengukuran Temperatur

* Resolusi pengukuran: 16 Bit
* Repeatability: ±0.2℃
* Range: At 25℃ ±2℃
* Waktu Respon: 1 / e (63%) 10 detik

Karakteristik Electrikal

* Power supply: DC 3.5 – 5.5V
* Konsumsi arus: measurement 0.3mA, standby 60μ A
* Periode sampling : lebih dari 2 detik



Gambar. Bentuk fisik DHT11

## Wiring Diagram

1) Untuk mengintegrasikan sensor DHT11 yang sudah terpasang pada Dev Shield dengan ESP32 maka gunakan program dibawah ini. (Ket : DHT11 terkoneksi ke pin 14 di Dev Shield)

## Upload sketch program via USB

1. Buat lembar kerja baru, kemudian ketikan program dibawah ini :

#include "DHT.h"

#define DHTPIN 14 // Digital pin connected to the DHT sensor

#define DHTTYPE DHT11 // DHT 11

DHT dht(DHTPIN, DHTTYPE);

void setup() {

Serial.begin(115200);

Serial.println(F("DHTxx test!"));

dht.begin();

}

void loop() {

delay(2000);

float h = dht.readHumidity();

float t = dht.readTemperature();

float f = dht.readTemperature(true);

if (isnan(h) || isnan(t) || isnan(f)) {

Serial.println(F("Failed to read from DHT sensor!"));

return;

}

float hif = dht.computeHeatIndex(f, h);

float hic = dht.computeHeatIndex(t, h, false);

Serial.print(F("Humidity: "));

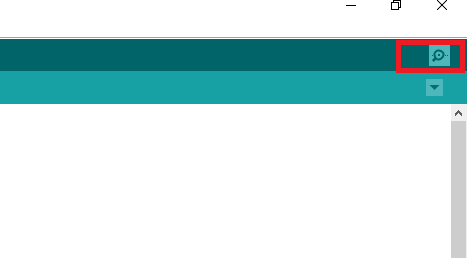
Serial.print(h);

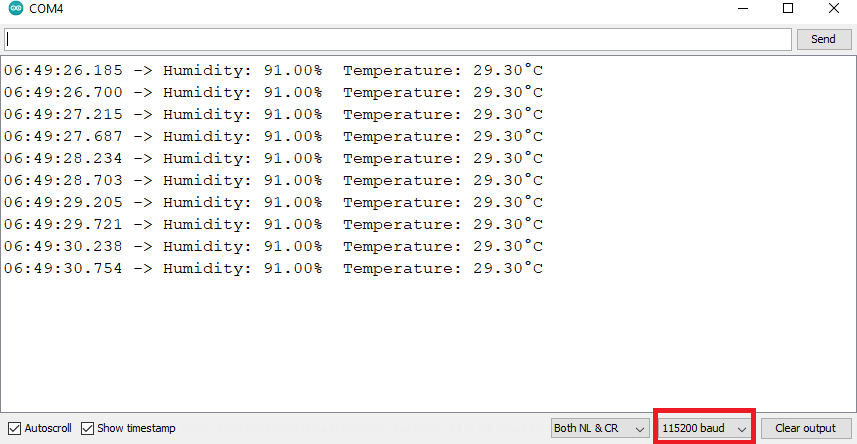
Serial.print(F("% Temperature: "));

Serial.print(t);

Serial.println (F("°C "));

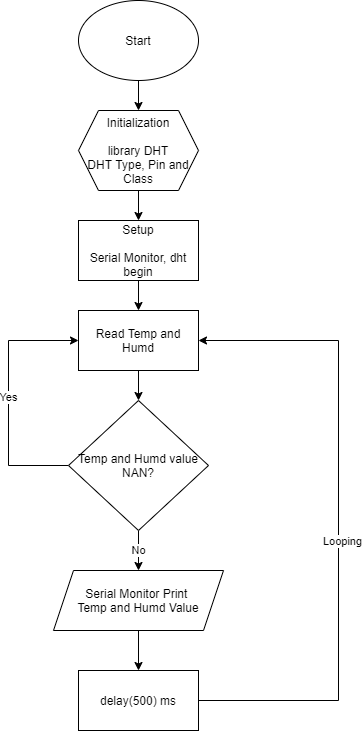
}

1. Simpan (Save) kemudian Upload. Tunggu proses compiling selesai dan sukses (tidak ada erorr). Sambungkan kabel data USB dari ESP32 dan Komputer/Laptop, kemudian tunggu proses uploading selesai dan sukses (mengisikan program dari komputer ke memori ESP32).
2. Setelah selesai kemudian buka serial monitor dengan cara menekan icon yang berada pada pojok kanan atas seperti pada gambar berikut
3. Pastikan kecepatan baudrate sesuai dengan program yaitu 115200 bps, maka akan tampil nilai suhu dan kelembaban.



* 1. **Penjelasan Program**

Berikut merupakan flowchart dari program Integrasi Sensor DHT



Gambar Flowchart

#include "DHT.h"

#define DHTPIN 14 // Digital pin connected to the DHT sensor

#define DHTTYPE DHT11 // DHT 11

DHT dht(DHTPIN, DHTTYPE);

Pada bagian ini terdapat printah untuk memasukan library, penentuan type dht,

penentuan pin data dht yang terhubung ke pin Developer Shield dan penamaan class pada dht yang digunakan.

void setup()

{

Serial.begin(115200); Serial.println(F("DHTxx test!")); dht.begin();

}

Serial.begin(115200) merupakan inisialisasi untuk mengaktifkan komunikasi serial

pada pin TX0 dan RX0 yaitu pin yang terhubung ke driver CP2023 untuk komunikasi serial dengan computer dengan kecepatan boudrate 115200 bps. Lalu dht.begin() merupakan perintah inisialisasi awal sensor dht.

float h = dht.readHumidity();

float t = dht.readTemperature();

float f = dht.readTemperature(true);

Pada bagian ini merupakan perintah untuk mendapatkan nilai pembacaan

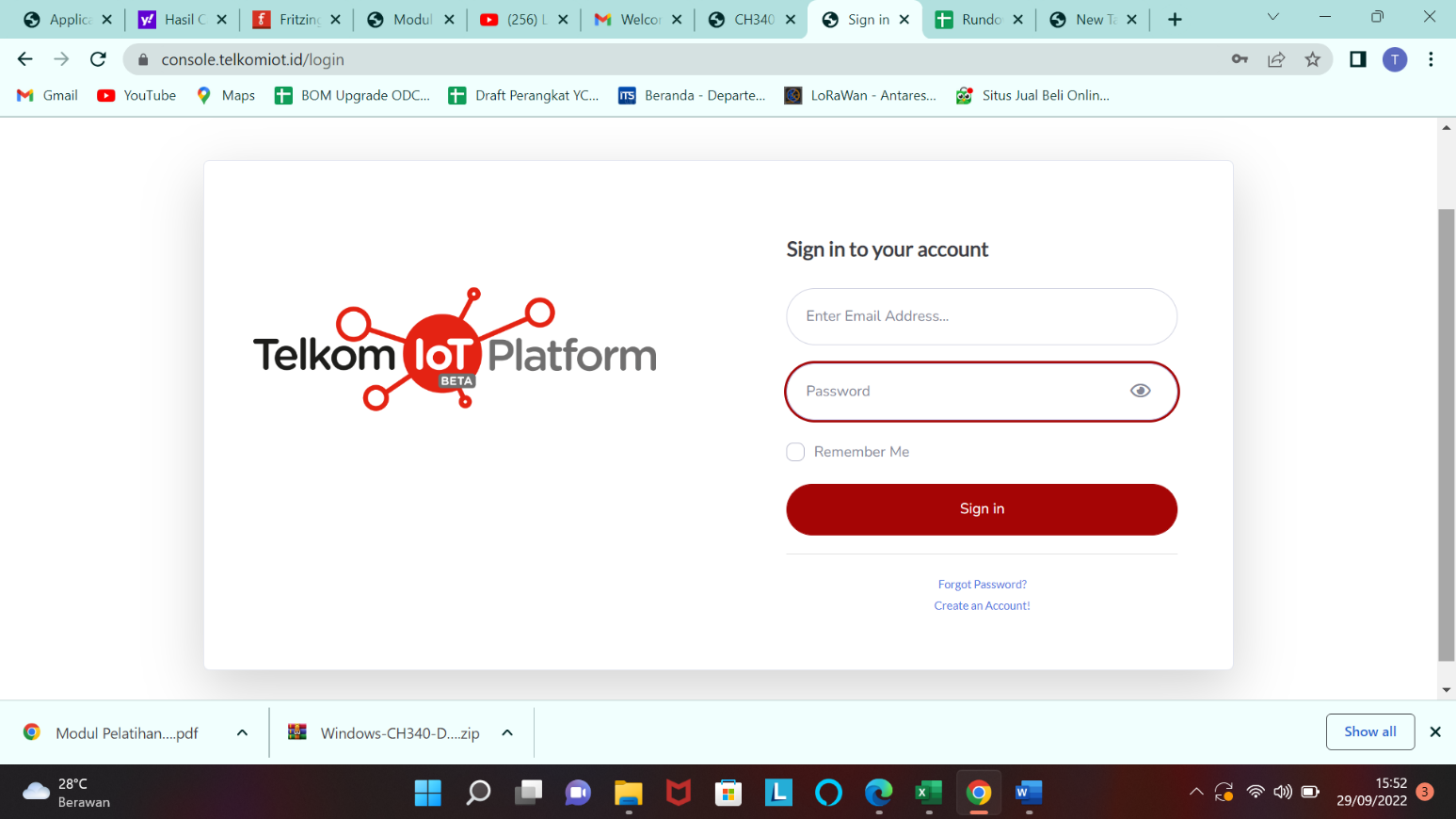
kelambaban dan suhu.

## IoT Platform

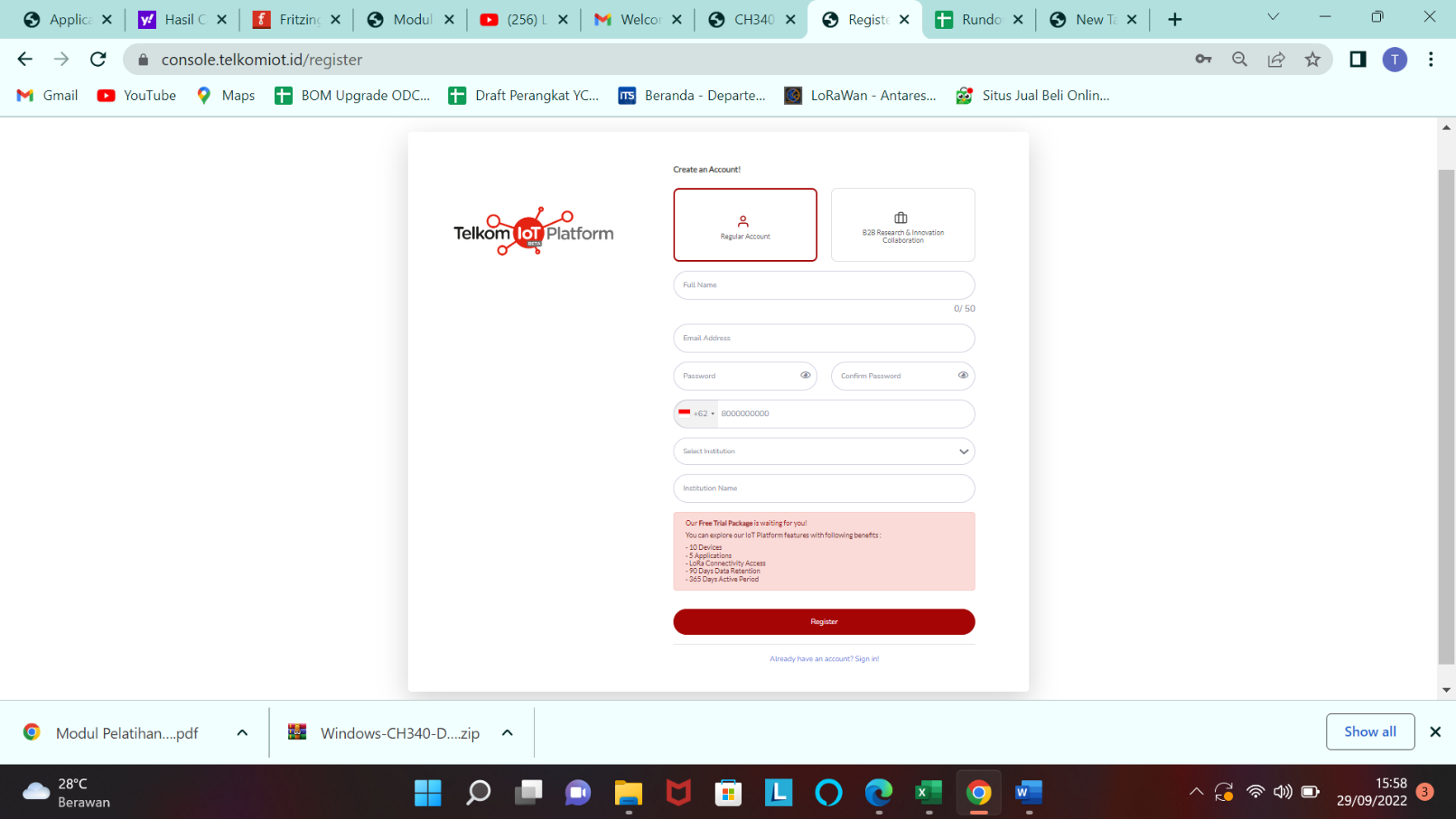
## Telkom IoT Platform adalah layanan teknologi IoT yang dapat dengan mudah menghubungkan, mengelola, mengotomatisasi berbagai perangkat atau sensor, dan memvisualisasikan data perangkat yang tersimpan di dashboard pribadi yang dapat disesuaikan dengan kebutuhan pelanggan. Dengan model bisnis Research, Innovation, and Collaboration, Telkom IoT Platform memiliki cakupan target pasar yang luas seperti akademisi, B2B hingga institusi besar sebagai pengguna.

# Register Akun

Buka web browser dan akses <https://console.telkomiot.id/>, klik menu create an account



Tampilan Halaman Login

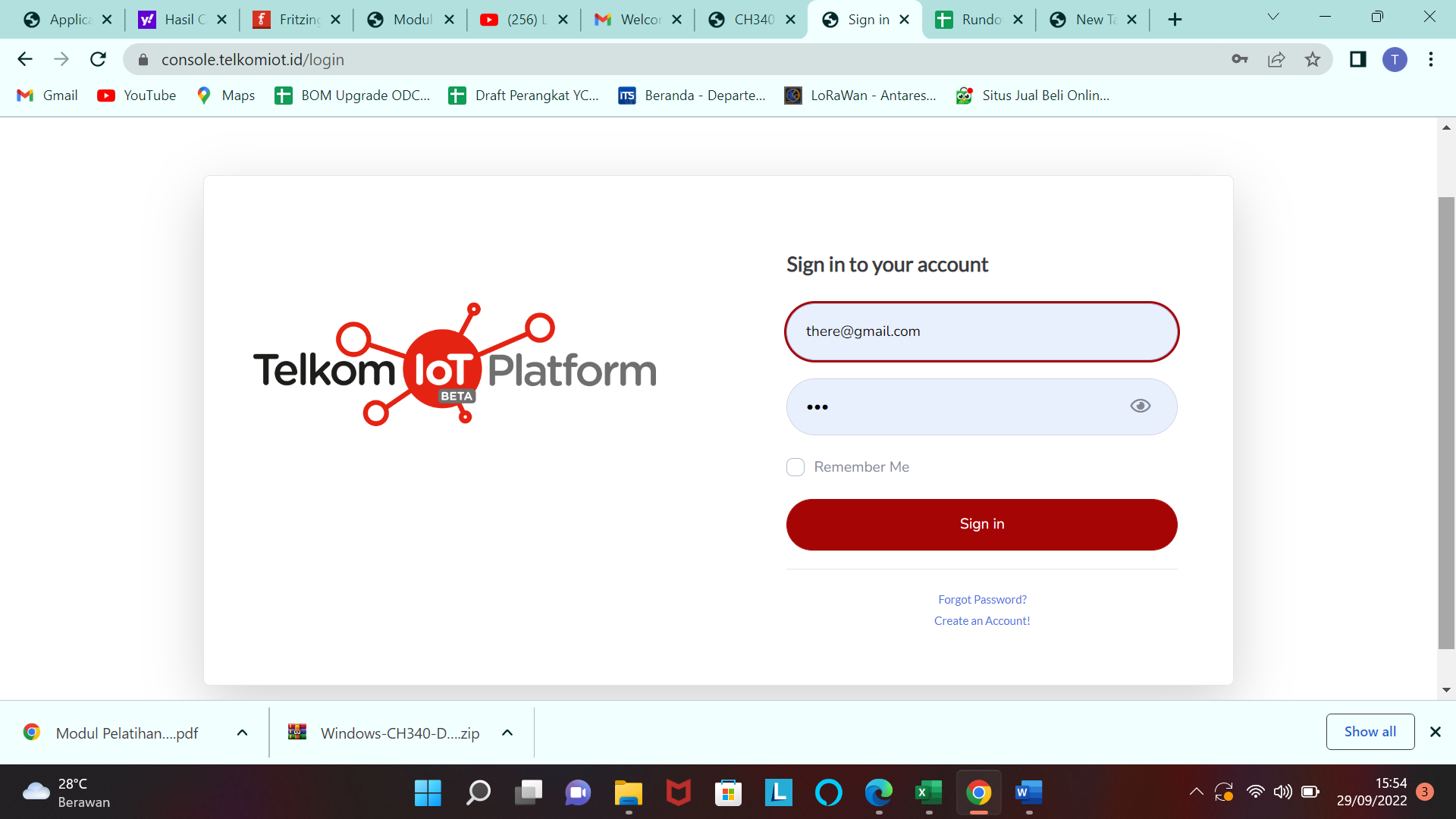


Halaman Create an Account

Pilih Regular Account kemudian Input data Fullname, Email Address, Password, Phone Number, Select Institution, Institution Name dan Klik Register. Buka email dari telkomiot.id dan verifikasi user

# Login Akun

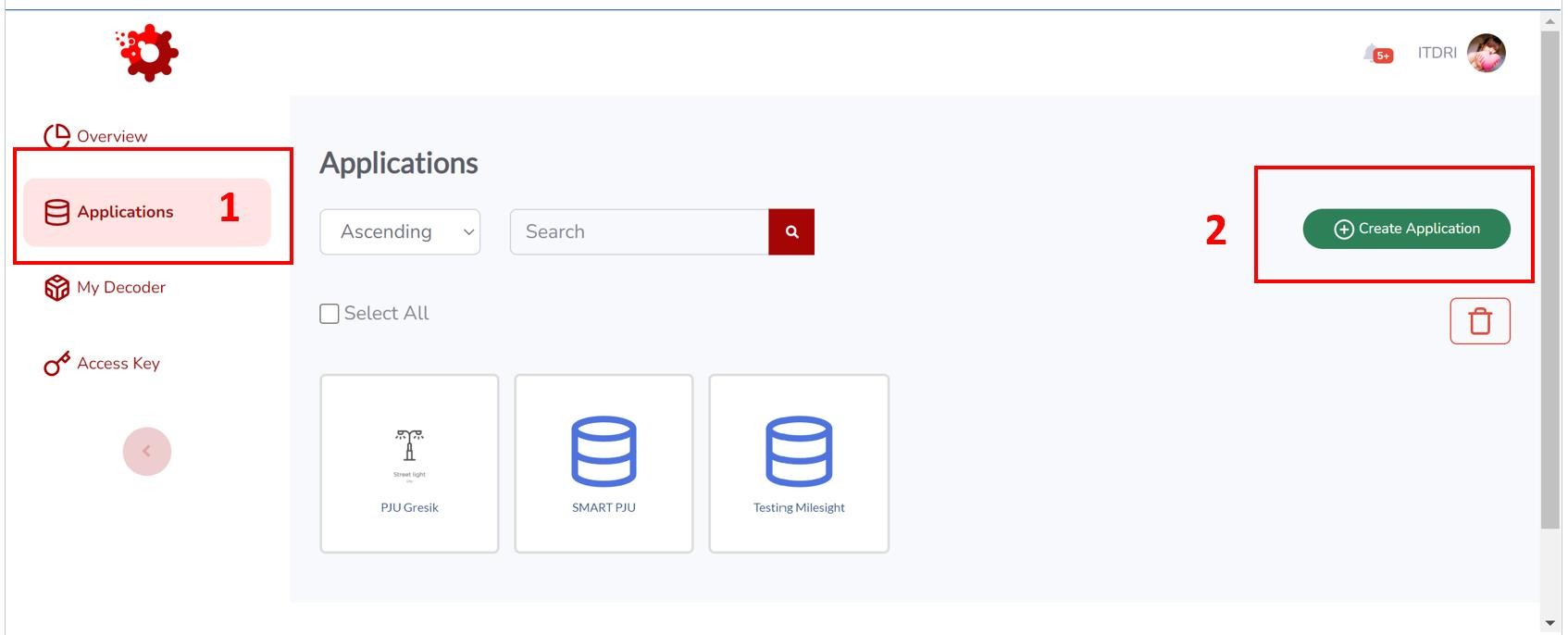
Buka web browser dan akses <https://console.telkomiot.id/>, isi username dan password yang sudah diregister dan klik tombol Sign In



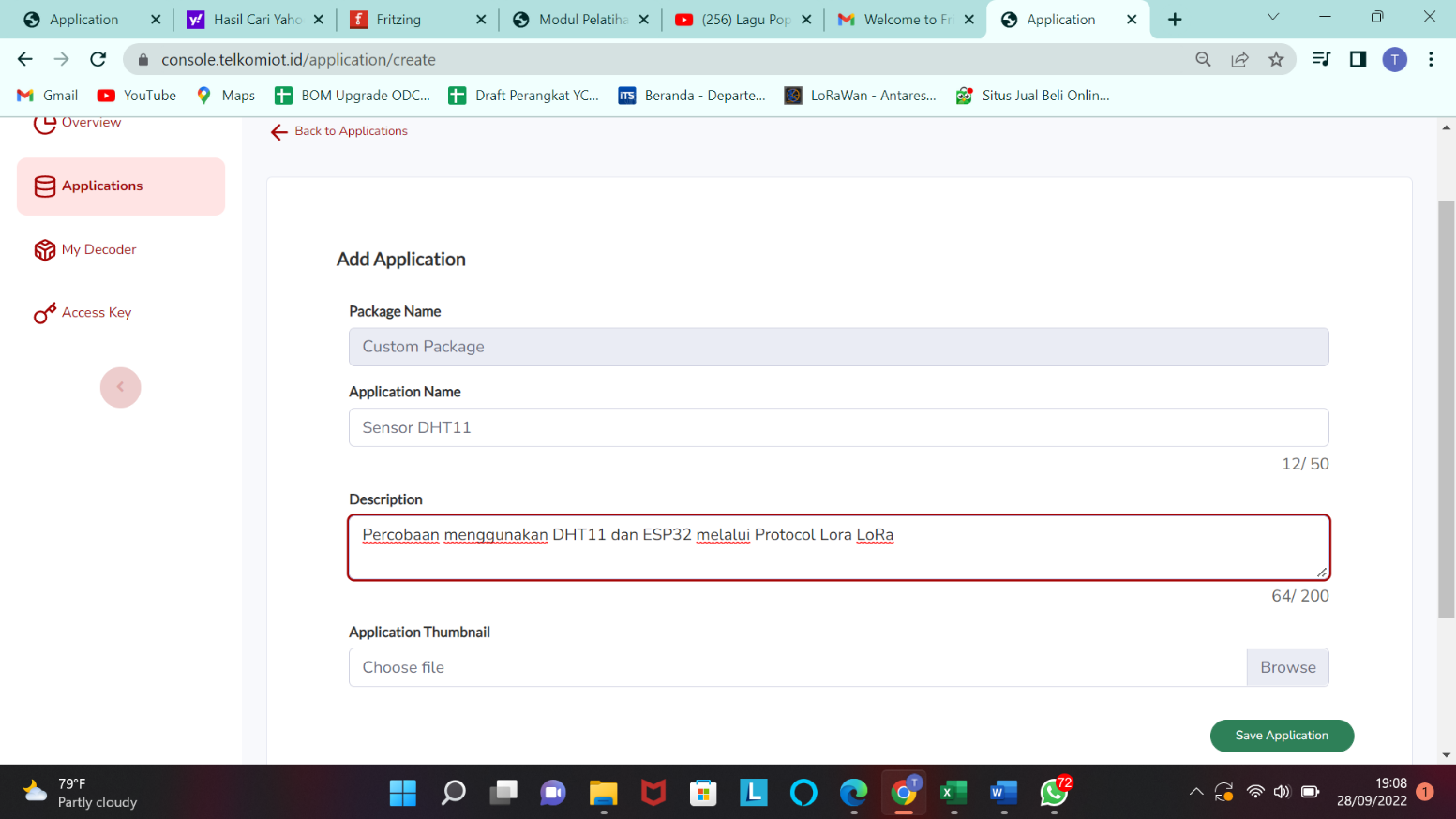
Halaman login

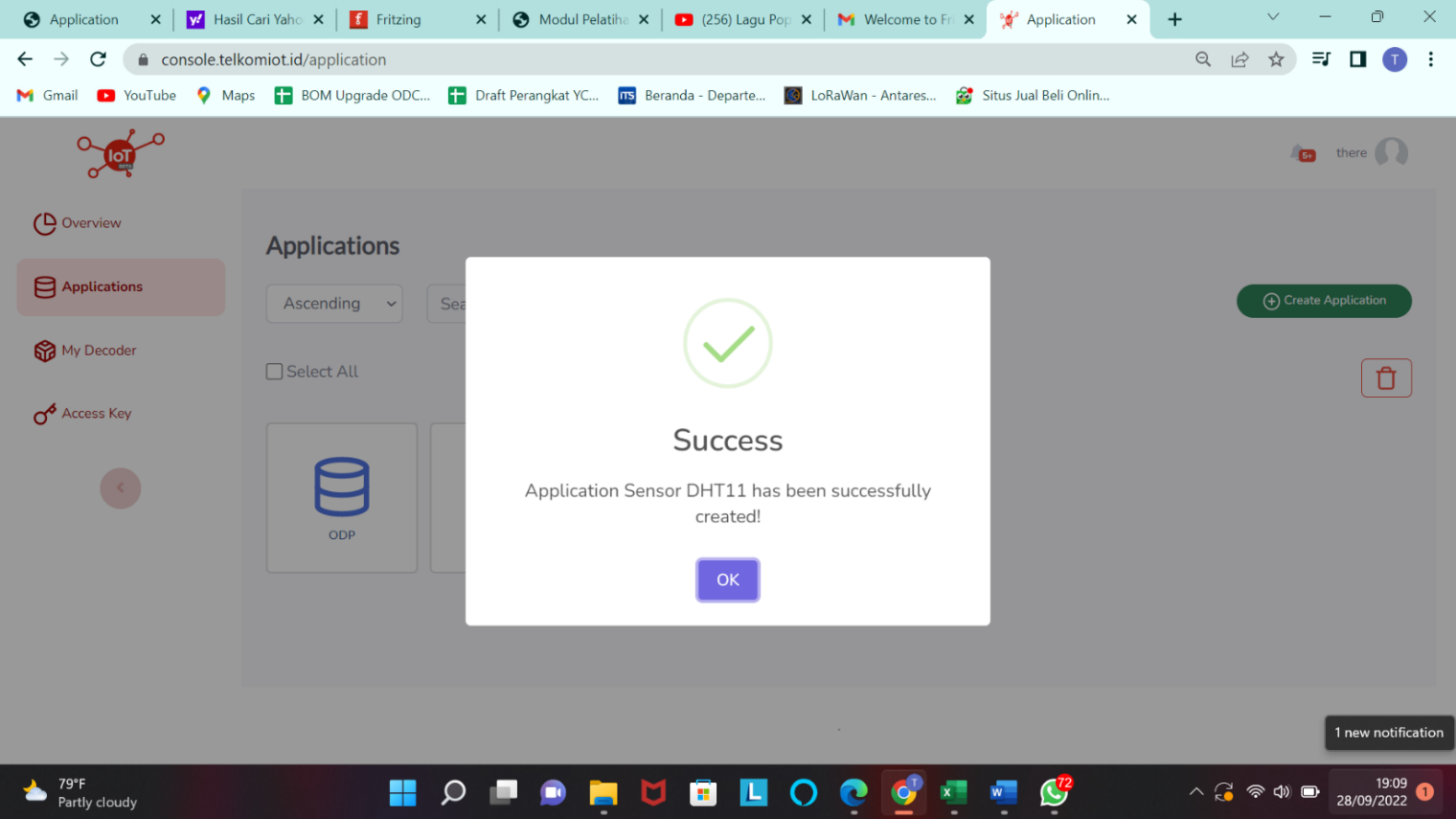
## Create Application

Buka web browser dan akses <https://console.telkomiot.id/>, klik menu application

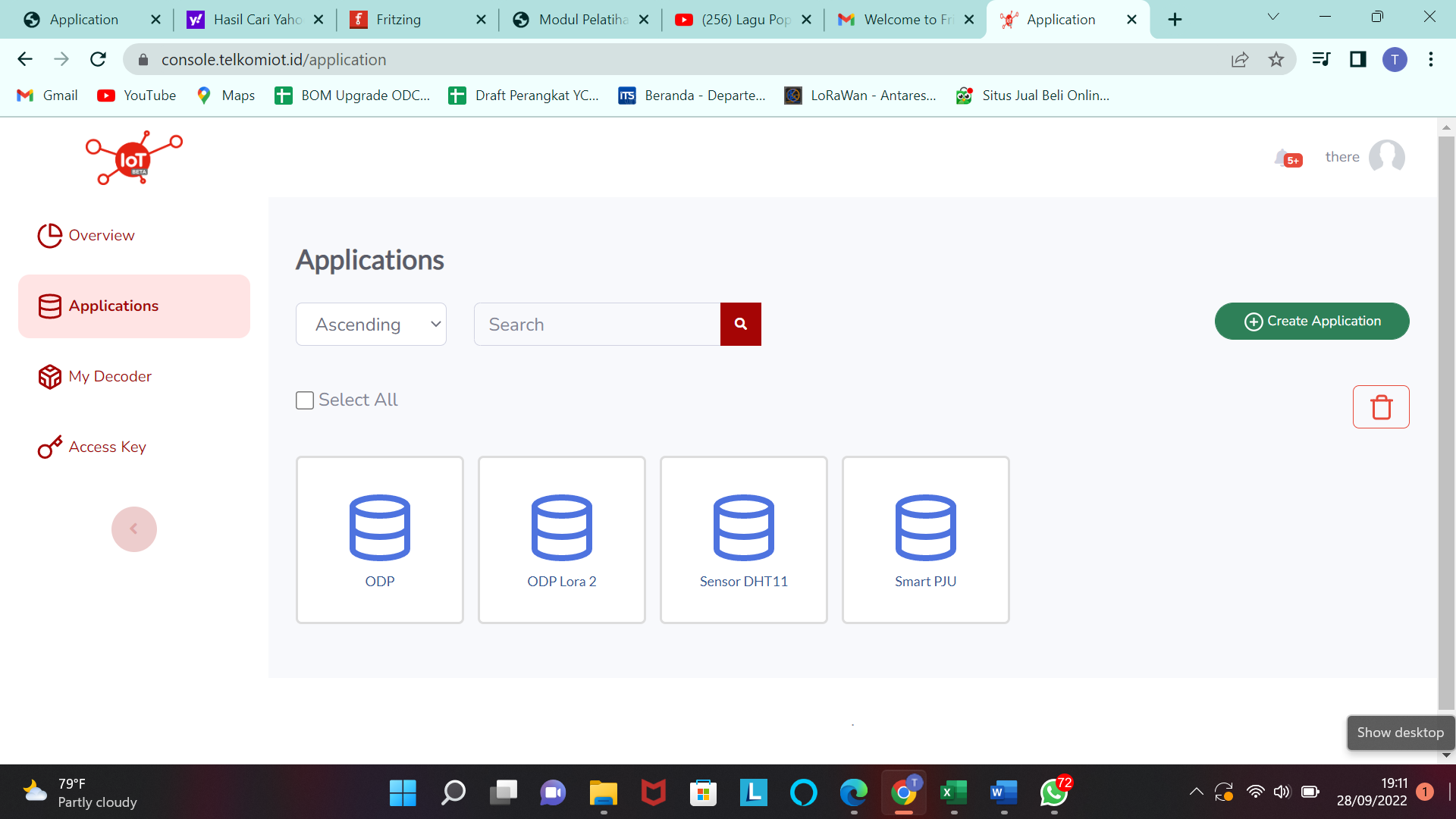


Halaman Application

Setelah masuk halaman Aplikasi, klik tombol Create Application

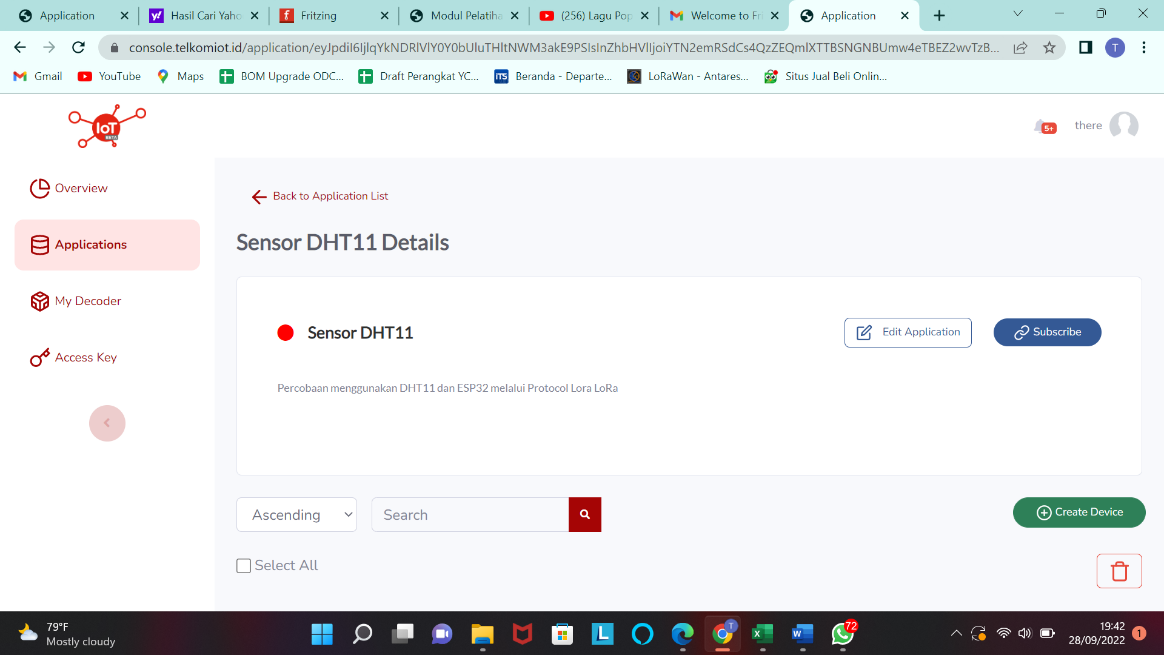
Isi parameter Application name, Description, Application Thumbnail dan klik tombol Save Application. Jika sudah maka akan muncul notifikasi dan tampilan halaman aplikasi sebagai Berikut

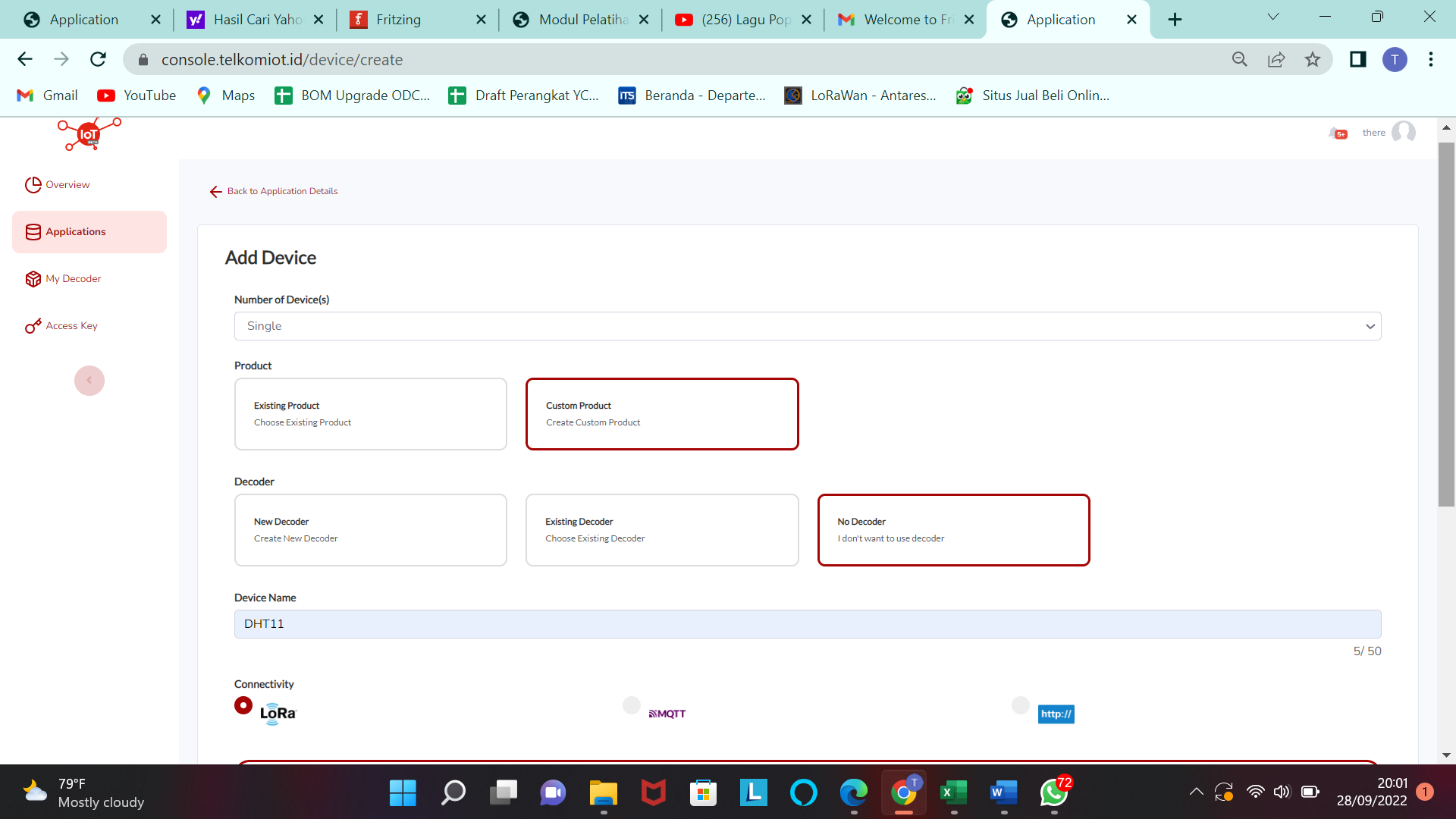
Jika telah berhasil akan menampilkan Alert Success



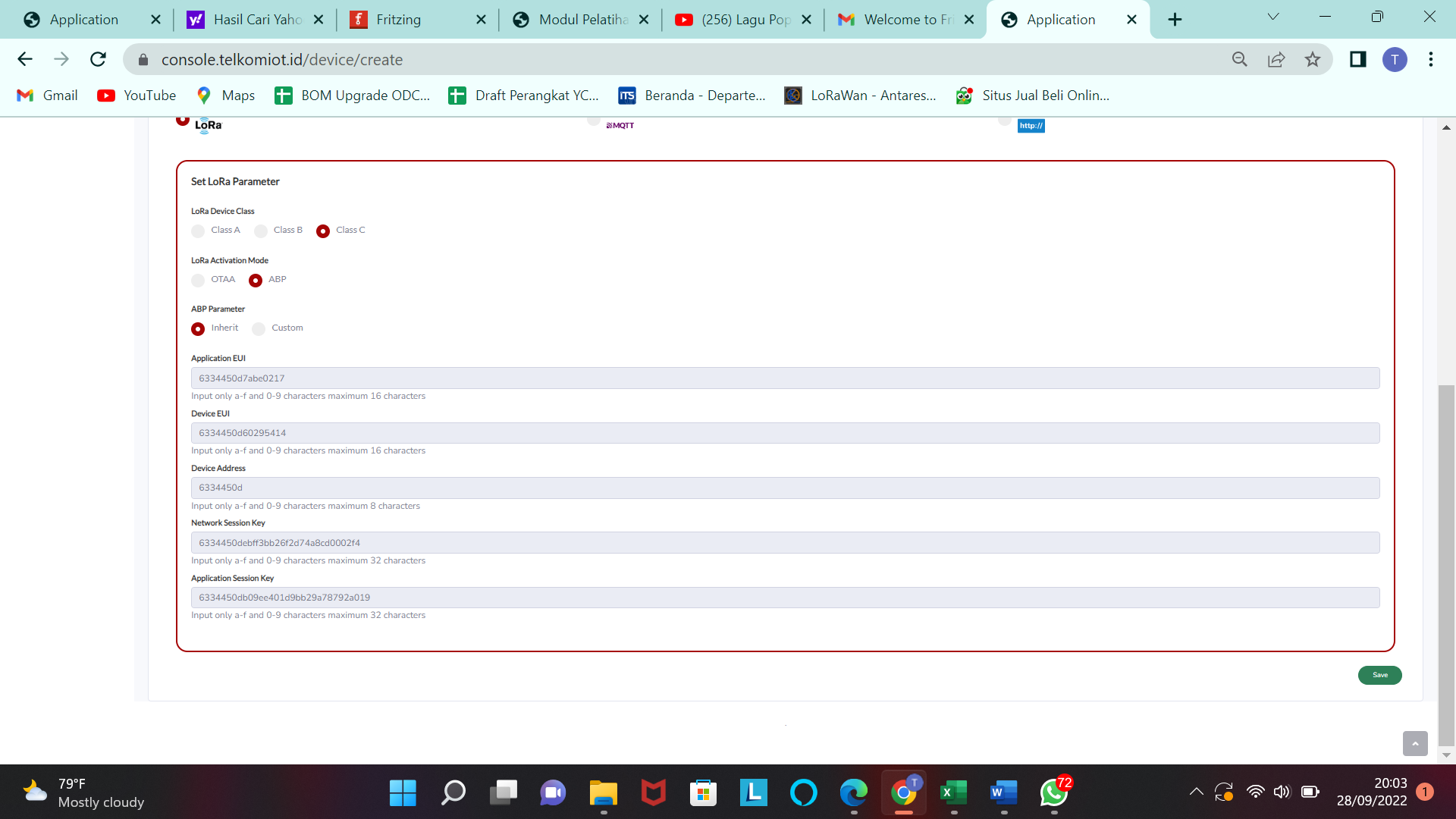
Halaman aplikasi baru

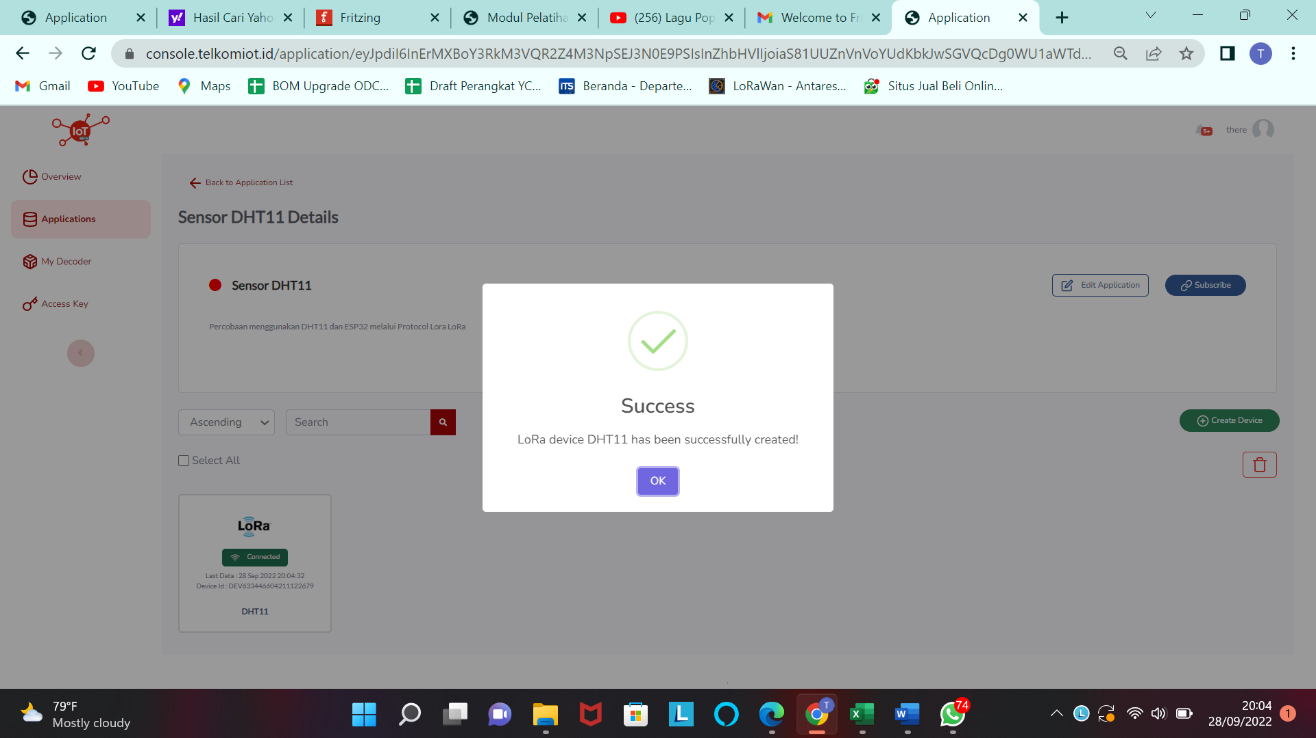
## Create Device

Buka halaman Application (<https://console.telkomiot.id/application>) menggunakan web browser dan klik aplikasi yang baru dibuat.



Dihalaman Add Device, isi parameter Number of Device (single), Product (Custom product), Decoder (No Decoder), Device Name, Connectivity (LoRa)

 Lalu Set LoRa Parameter seperti gambar diatas kemudian klik Save (Simpan)

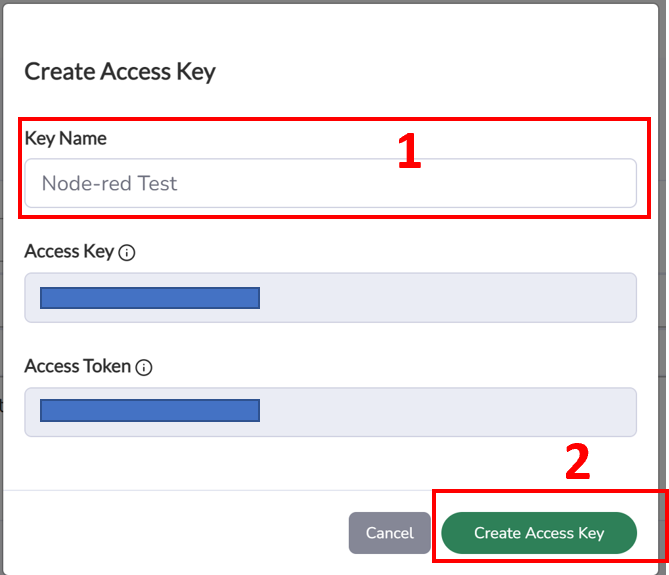
 Setelah Berhasil maka akan tampil Alert dibawah ini.

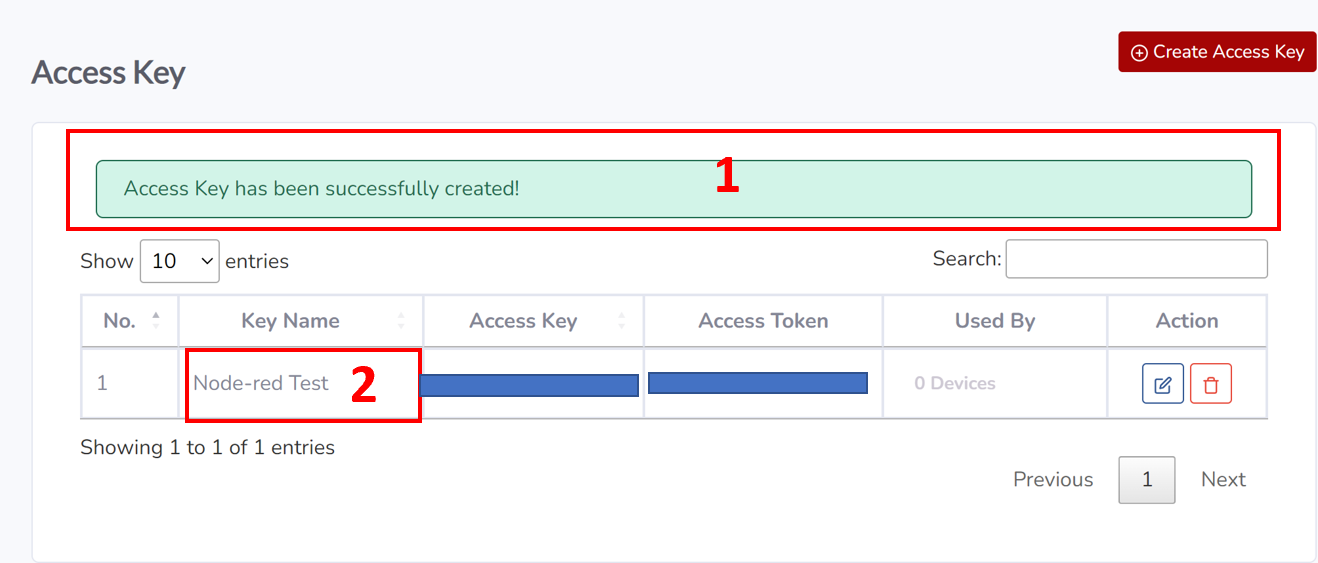
## Create Access Key dan Access Token

Buka halaman Application (https://console.telkomiot.id/access-key) menggunakan web browser dan klik Create Access Keys.



Isi nama token sesuai dengan yang diinginkan seperti pada gambar dibawah ini, lalu klik Creat e Access Key ,



 Jika sudah selesai di buat, maka tampilan access keys seperti gambar berikut

**6. Integrasi Device ke Telkom IoT Platform melalui Protokol LoRa**

**a. Arsitektur LoRaWan**

Gambar Arsitektur LoRaWan

LoRaWAN dibangun menggunakan topologi 'star to star' yang memungkinkan device dapat bekerja menggunakan baterai dalam jangka waktu yang lebih lama dibandingkan topologi mesh network pada umumnya. Pada arsitektur LoRaWAN, device tidak ter-asosiasi dengan gateway tertentu, nantinya data dari device akan di terima oleh beberapa gateway dalam jangkauan jaringan LoRa. Tiap gateway akan meneruskan paket yang diterima dari device ke network server yang ada di cloud melalui backhaul seperti ethernet, wifi, satelit ataupun seluler. Pemrosesan dilakukan di level network server yang mengatur jaringan, memfilter paket yang di terima, security check, penjadwalan ACK (acknowledments), dan mengatur adaptive data rate (ADR). Ketika device bergerak tidak diperlukan handover antar gateway, ini merupakan fiur utama untuk menjamin aplikasi LoRaWAN seperti asset tracking yang memang menjadi salah satu target utama vertikal IoT LoRaWAN.

1. **Upload Sketch Program**
2. Buat lembar kerja baru, kemudian ketikan program dibawah ini :

#include <lorawan.h>

#include <DHT.h>

#define DHTPIN 14

const char \*devAddr = "cdb32373"; // Ganti dengan device adrress masing masing

const char \*appSKey = "033ee0911aea460fc60e05f0d78fc3b7"; // Ganti dengan Apps Key masing masing

const char \*nwkSKey = "148a91925d191edc1a0b21f672ab57ed"; // Ganti dengan Nwks Key masing masing

const unsigned long interval = 10000; // interval pengiriman data 10 detik

unsigned long previousMillis = 0;

#define DHTTYPE DHT11 // tipe DHT

DHT dht(DHTPIN, DHTTYPE);

char myStr[50];

byte outStr[255];

int port, channel, freq;

#define MSG\_BUFFER\_SIZE (150)

char msg[MSG\_BUFFER\_SIZE] = "{\"t\":\"%.2f\",\"h\":\"%.2f\"}";

char pesan[150];

const sRFM\_pins RFM\_pins = {

.CS = 5,

.RST = 0,

.DIO0 = 27,

.DIO1 = 2,

};

void setup() {

Serial.begin(115200);

delay(2000);

if (!lora.init()) {

Serial.println("RFM95 not detected");

delay(5000);

return;

}

dht.begin();

lora.setDeviceClass(CLASS\_C);

lora.setDataRate(SF10BW125);

lora.setFramePortTx(5);

lora.setChannel(MULTI);

lora.setTxPower(15);

lora.setNwkSKey(nwkSKey);

lora.setAppSKey(appSKey);

lora.setDevAddr(devAddr);

}

void loop() {

if (millis() - previousMillis > interval) {

previousMillis = millis();

float h = dht.readHumidity();

float t = dht.readTemperature();

if (isnan(h) || isnan(t)) {

h = 0;

t = 0;

}

sprintf(myStr, msg,t,h );

Serial.print("Sending: ");

Serial.println(myStr);

lora.sendUplink(myStr, strlen(myStr), 0);

port = lora.getFramePortTx();

channel = lora.getChannel();

freq = lora.getChannelFreq(channel);

Serial.print(F("fport: ")); Serial.print(port); Serial.print(" ");

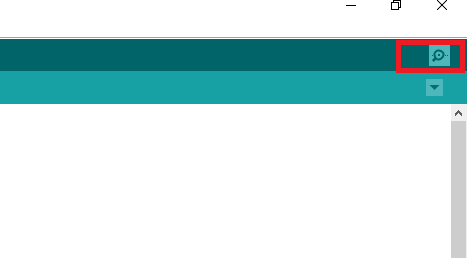
Serial.print(F("Ch: ")); Serial.print(channel); Serial.print(" ");

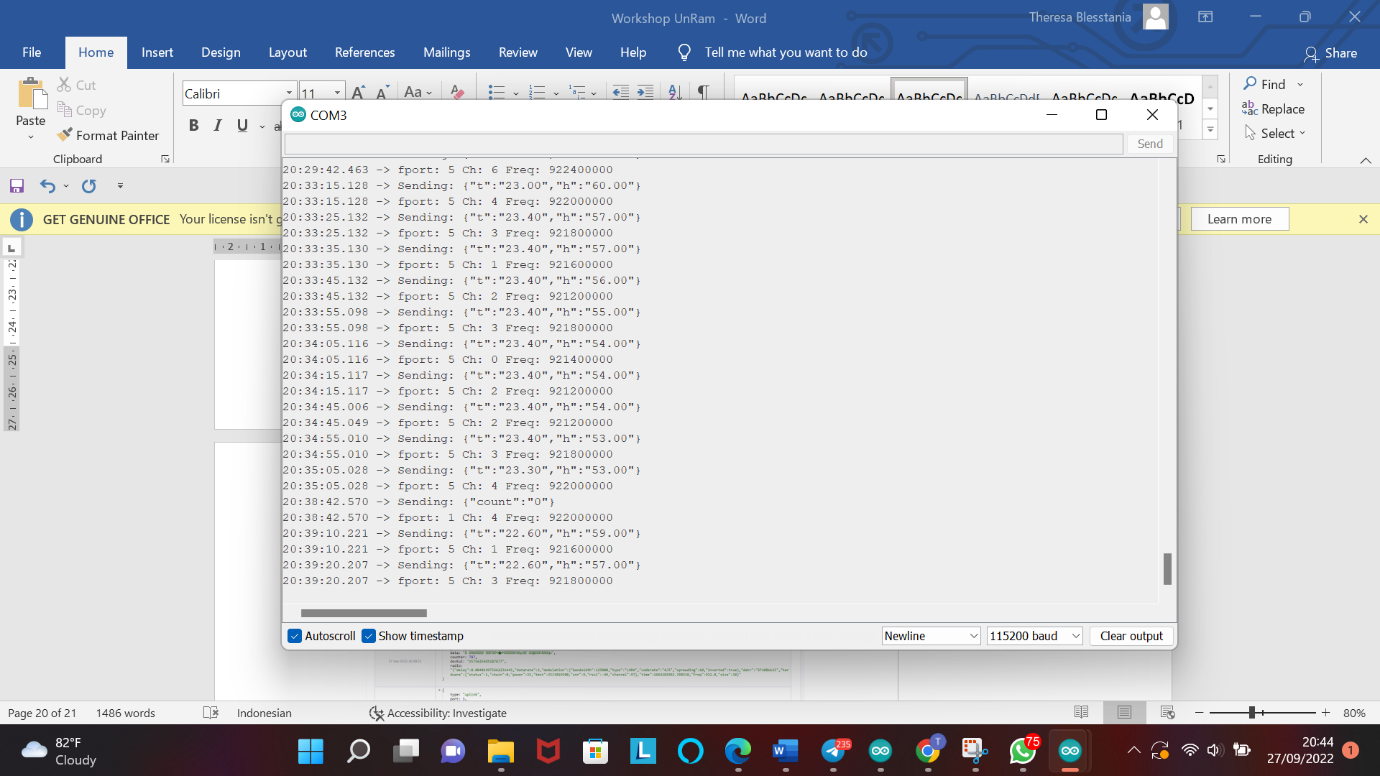
Serial.print(F("Freq: ")); Serial.print(freq); Serial.println(" ");

}

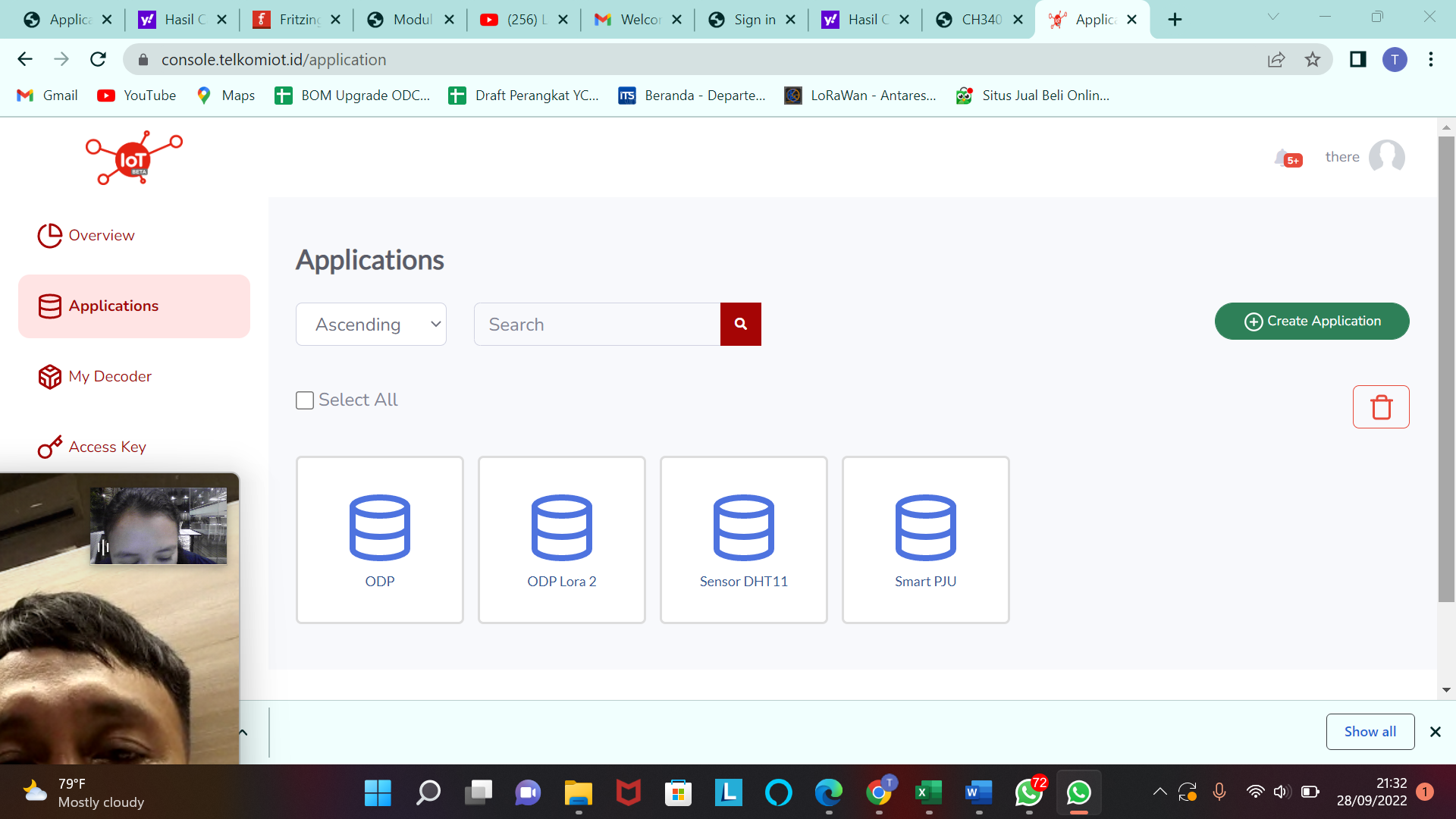
lora.update();

}

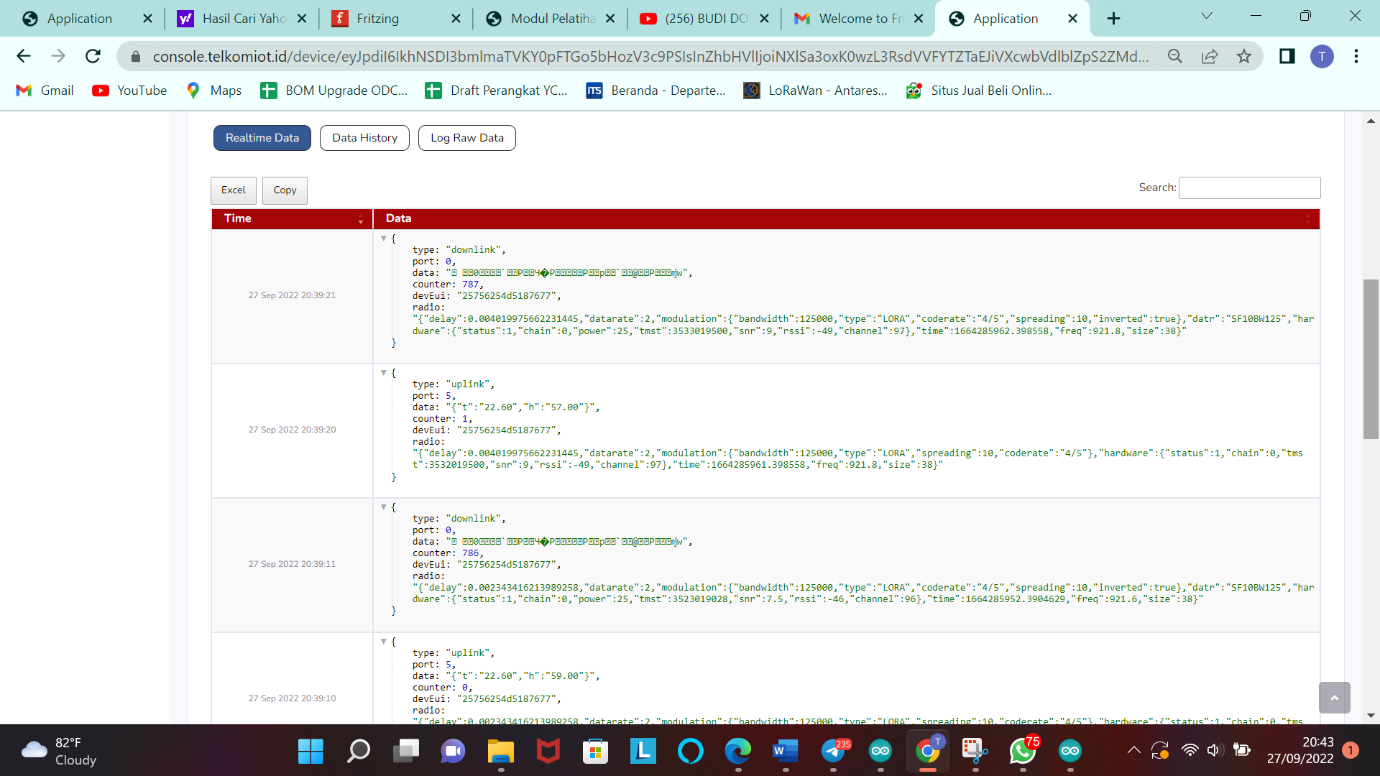
1. Simpan (Save) kemudian Upload. Tunggu proses compiling selesai dan sukses (tidak ada erorr). Sambungkan kabel data USB dari ESP32 dan Komputer/Laptop, kemudian tunggu proses uploading selesai dan sukses (mengisikan program dari komputer ke memori ESP32).
2. Setelah selesai kemudian buka serial monitor dengan cara menekan icon yang berada pada pojok kanan atas seperti pada gambar berikut
3. Pastikan kecepatan baudrate sesuai dengan program yaitu 115200 bps, maka akan tampil nilai suhu dan kelembaban.



1. **Monitoring data Real Time diDashboard Telkom IoT Platform**
   * 1. Untuk melihat data yang masuk ke Telkom IoT Platform dapat masuk kembali ke Dashboard Telkom IoT Platform kemudian masuk ke menu Application>Kemudian klik pada device yang sudah dibuat.

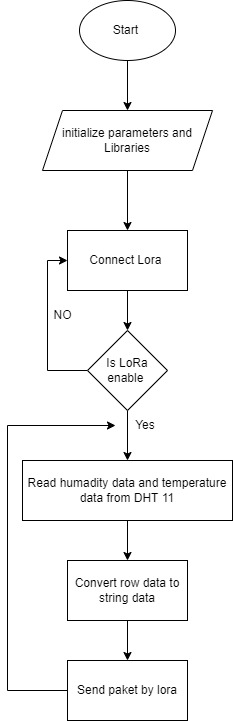


* + 1. Maka pada menu *Realtime Data* akan menampilkan data yang dikirim dari Device .



1. **Keterangan Program**

Berikut merupakan flowchart dari program Integrasi Sensor DHT ke Telkom IoT Platform by LoRa

****

Gambar Flowchart

Program dibawah ini untuk untuk mengatur koneksi ke Telkom IoT Platform dengan mengisi Device Address,Network Session Key dan Application session key yang didapatkan dari Dashboard Telkom IoT Platform.

const char \*devAddr = "cdb32373"; // Ganti dengan device adrress masing masing

const char \*appSKey = "033ee0911aea460fc60e05f0d78fc3b7"; // Ganti dengan Apps Key masing masing

const char \*nwkSKey = "148a91925d191edc1a0b21f672ab57ed"; // Ganti dengan Nwks Key masing masing

Data yang dikirimkan ke Telkom IoT Platform harus dalam format JSON, maka program ini berfungsi agar data sensor dikonversi dalam bentuk JSON

#define MSG\_BUFFER\_SIZE (150)

char msg[MSG\_BUFFER\_SIZE] = "{\"t\":\"%.2f\",\"h\":\"%.2f\"}";

char pesan[150];

Inisialisasi Pin pada modul dari Modul LoRa yang terhbung ke ESP32

const sRFM\_pins RFM\_pins = {

.CS = 5,

.RST = 0,

.DIO0 = 27,

.DIO1 = 2,

};

Agar LoRa dapat terkoneksi maka dibutuhkan untuk mengatur Class, Data Rate,Frame Port, Channel, TxPower, Network Session Key, Application Session Key dan Device Address

lora.setDeviceClass(CLASS\_C);

lora.setDataRate(SF10BW125);

lora.setFramePortTx(5);

lora.setChannel(MULTI);

lora.setTxPower(15);

lora.setNwkSKey(nwkSKey);

lora.setAppSKey(appSKey);

lora.setDevAddr(devAddr);

Bagian ini untuk menampilkan data sensor baik pada serial monitor maupun pada dashboard Telkom IoT Platform

sprintf(myStr, msg,t,h );

Serial.print("Sending: ");

Serial.println(myStr);

lora.sendUplink(myStr, strlen(myStr), 0);

port = lora.getFramePortTx();

channel = lora.getChannel();

freq = lora.getChannelFreq(channel);

Serial.print(F("fport: ")); Serial.print(port); Serial.print(" ");

Serial.print(F("Ch: ")); Serial.print(channel); Serial.print(" ");

Serial.print(F("Freq: ")); Serial.print(freq); Serial.println(" ");

**7. Referensi**

* Arduino IDE: [*https://www.arduino.cc/en/software*](https://www.arduino.cc/en/software)
* Preference Board ESP32: [*https://dl.espressif.com/dl/package\_esp32\_index.json*](https://dl.espressif.com/dl/package_esp32_index.json)
* Dashboard Telkom IoT Platform: [*https://console.telkomiot.id/*](https://console.telkomiot.id/)
* Driver USB CH340: [*https://www.drduino.com/blogs/news/ch340-usb-drivers*](https://www.drduino.com/blogs/news/ch340-usb-drivers)
* Pembelian Development Shield ESP32 & Sensor: <https://tokopedia.link/GAIYtT0pItb>