

LAMPIRAN

Lampiran 1. Code Node 1

Berikut ditampilkan bagian-bagian kode program pada Node 1 yang merepresentasikan logika utama dan kontribusi teknis dari sistem yang dibangun. Kode secara keseluruhan tidak ditampilkan untuk mencegah duplikasi/penyalahgunaan; kredensial (*SSID*, *password*, *API key*, dan *device ID*) telah disamarkan.

1. Konversi Nilai ADC ke PPM

```
int convertToPPM(int adcValue, int baseline)
{
    int ppm = map(adcValue, baseline, 4095, 0, 500);
    return ppm < 0 ? 0 : ppm;
}
```

Fungsi ini mengonversi nilai pembacaan mentah ADC dari sensor MQ-4 menjadi satuan konsentrasi gas (PPM), menggunakan nilai baseline hasil kalibrasi pada kondisi udara bersih sebagai titik referensi bawah, dan nilai maksimum ADC (4095, resolusi 12-bit ESP32) sebagai titik referensi atas.

2. Logika Klasifikasi Status Gas dan Suhu

```
String statusGas = (ppm1 > batasPPM || ppm2 > batasPPM) ?
"BAHAYA_GAS" : "GAS_AMAN";

String statusSuhu = (suhu > batasSuhu) ? "SUHU_TINGGI" :
"SUHU_AMAN";
```

Bagian ini merupakan inti pengambilan keputusan sistem berdasarkan ambang batas (*threshold*) yang telah ditentukan. Jika salah satu dari dua sensor MQ-4 melebihi

batas PPM, status akan berubah menjadi kondisi bahaya; demikian pula untuk suhu terhadap batas maksimum yang ditentukan.

3. Penyusunan String-to-Sign untuk Autentikasi Tuya Cloud API

```
String stringToSign = method + "\n" + bodyHash + "\n" + "\n" +
path;

String signStr;
if (accessToken == "")
    signStr = String(TUYA_CLIENT_ID) + String(t) + nonce +
stringToSign;
else
    signStr = String(TUYA_CLIENT_ID) + accessToken + String(t) +
nonce + stringToSign;

// signStr kemudian di-hash menggunakan HMAC-SHA256 dengan client
secret
// menghasilkan signature yang dikirim pada header "sign"
```

Potongan ini menunjukkan pemahaman terhadap skema autentikasi Tuya Cloud API, yaitu penyusunan string tanda tangan (*string-to-sign*) dari kombinasi *client ID*, *access token*, *timestamp*, *nonce*, dan *hash body request*, sebelum ditandatangani menggunakan algoritma HMAC-SHA256. Implementasi lengkap fungsi hashing tidak ditampilkan karena merupakan algoritma kriptografi standar, bukan kontribusi orisinal penelitian.

4. Struktur Payload Data ke Tuya Cloud

```
JsonObject properties = bodyDoc.createNestedObject("properties");
properties["Suhu"]      = (int)(suhuVal * 10);    // skala x10
properties["gas_mq4_1"] = ppm1Val;
properties["gas_mq4_2"] = ppm2Val;
properties["status_gas"] = statusGas;
```

Struktur ini menunjukkan skema data (payload) yang dikirimkan dari perangkat ke *dashboard* Tuya melalui protokol HTTP, merepresentasikan rancangan komunikasi antara node sensor dan *platform cloud*.

5. Struktur Topik MQTT (HiveMQ)

```
hivemq.publish("kandang/mq4_1", String(ppm1).c_str());
hivemq.publish("kandang/mq4_2", String(ppm2).c_str());
hivemq.publish("kandang/suhu", String(suhu, 1).c_str());
hivemq.publish("kandang/humidity", String(humidity, 1).c_str());
hivemq.publish("kandang/status_gas", statusGas.c_str(), true);
hivemq.publish("kandang/status_suhu", statusSuhu.c_str(), true);
```

Bagian ini memperlihatkan struktur penamaan topik (topic naming convention) yang digunakan pada *broker* MQTT HiveMQ, sebagai bagian dari arsitektur komunikasi data *real-time* pada sistem.

Lampiran 2. Code Node 2

Node 2 menggunakan sensor flow YF-S201 untuk mengukur volume air yang mengalir. Skema autentikasi Tuya Cloud API (fungsi *buildSign*, *sha256Hash*, *hmacSHA256*) identik dengan Node 1, Bagian yang ditampilkan berikut adalah logika yang spesifik/unik untuk Node 2.

1. Interrupt Service Routine (ISR) Sensor Flow

```
void IRAM_ATTR flowISR()
{
    pulseCount++;
}

// pada setup():
pinMode(FLOW_SENSOR_PIN, INPUT_PULLUP);
attachInterrupt(digitalPinToInterrupt(FLOW_SENSOR_PIN), flowISR,
FALLING);
```

Sensor YF-S201 menghasilkan pulsa setiap satuan volume air yang melewatinya. Karena pulsa terjadi sangat cepat, pembacaan dilakukan menggunakan *hardware* interrupt (bukan polling) agar tidak ada pulsa yang terlewat, dengan *pulseCount* bertipe *volatile* agar aman diakses di dalam ISR.

2. Konversi Pulsa ke Volume (Kalibrasi)

```
const float ML_PER_PULSE = 1.8664; // calibration factor 8.93,
pulsa/liter = 535.8

unsigned long pulseBaru = pulseCount - lastPulseCount;
lastPulseCount = pulseCount;
```

```

if (pulseBaru > 0) {
    volumeML += pulseBaru * ML_PER_PULSE;
}

```

Nilai ML_PER_PULSE merupakan hasil kalibrasi sensor (volume per satu pulsa), yang menjadi dasar perhitungan volume kumulatif air (volumeML) dari jumlah pulsa yang terdeteksi.

3. Logika Deteksi Awal/Akhir Aliran Air



```

// Ada aliran air
if (pulseBaru > 0)
{ sedangMengalir = true;
  volumeTer kirim = false;
  lastFlowTime = millis();
}

// Air berhenti selama 3 detik -> kirim volume akhir
if (sedangMengalir && (millis() - lastFlowTime > 3000))
{ sedangMengalir = false;
  if (!volumeTer kirim) {
    // kirim volume akhir ke HiveMQ & Tuya
    // lalu reset volumeML, pulseCount = 0
    volumeTer kirim = true;
  }
}

```

Ini adalah bagian logika terpenting pada Node 2: sistem mendeteksi kapan aliran air dimulai dan kapan berhenti (menggunakan jeda waktu 3 detik tanpa pulsa baru sebagai penanda aliran berhenti), lalu mengirimkan volume akhir hanya satu kali (ditandai flag `volumeTer kirim`) sebelum melakukan reset. Pendekatan ini mencegah pengiriman data berulang untuk satu siklus pengisian air yang sama.

4. Struktur Payload Volume ke Tuya Cloud

```
DynamicJsonDocument bodyDoc(256);
JsonObject properties = bodyDoc.createNestedObject("properties");
properties["total_air"] = (int)volumeVal; // mL, integer
```

Payload yang dikirim ke Tuya untuk Node 2 hanya berisi satu parameter, yaitu total volume air (`total_air`) dalam mililiter, berbeda dengan Node 1 yang mengirim beberapa parameter sekaligus (suhu dan gas).

Lampiran 3. Code Node 3

Node 3 berperan sebagai aktuator: menerima data status dari Node 1 melalui MQTT (*subscriber*) dan mengendalikan relay/solenoid secara otomatis, dengan dukungan mode override manual dari aplikasi IoT MQTT Panel.

1. Logika Kendali Relay Otomatis dengan Override Manual.

```
void updateRelay()
{
    if (manualMode) return;    // mode manual: abaikan logika
                                otomatis
```

```

    if (statusBahayaGas || statusSuhuTinggi)
        relayON();
    else
        relayOFF();
}

```

Fungsi ini adalah inti kendali aktuator: relay akan aktif secara otomatis jika status bahaya gas atau suhu tinggi terdeteksi. Namun jika manualMode sedang aktif (pengguna mengendalikan lewat aplikasi MyMQTT), logika otomatis ini diabaikan sepenuhnya sehingga tidak ada konflik antara kontrol otomatis dan manual.

2. Callback MQTT 4 Penanganan Pesan Masuk

```

if (topicStr == "kandang/status_gas")
{
    statusBahayaGas = (message == "BAHAYA_GAS");
    updateRelay();
}
else if (topicStr == "kandang/status_suhu")
{
    statusSuhuTinggi = (message == "SUHU_TINGGI");
    updateRelay();
}
else if (topicStr == "kandang/manual_siram") {
    if (message == "ON") { manualMode = true; relayON(); }
    if (message == "OFF") { manualMode = false; updateRelay(); }
}

```

Fungsi callback ini merupakan penerapan pola event-driven berbasis MQTT: setiap topik yang diterima memperbarui variabel status terkait (gas, suhu, atau mode manual),

kemudian memicu evaluasi ulang kondisi relay melalui `updateRelay()`. Struktur ini menunjukkan bagaimana tiga sumber input (dua otomatis dari sensor, satu manual dari pengguna) diintegrasikan menjadi satu keputusan kendali yang konsisten.

3. Subscribe Topik saat Koneksi MQTT Terbentuk

```
client.subscribe("kandang/status_gas");
client.subscribe("kandang/status_suhu");
client.subscribe("kandang/manual_siram");
```

Bagian ini menunjukkan tiga topik yang menjadi masukan kendali Node 3, melengkapi gambaran arsitektur komunikasi MQTT secara menyeluruh antar-node (Node 1 sebagai publisher status, aplikasi MyMQTT sebagai kontrol manual, dan Node 3 sebagai subscriber/aktuator).

Lampiran 4. Dokumentasi Tahapan API Tuya

1. Tahap pertama adalah membuat akun pada Tuya IoT Platform sebagai media pengelolaan perangkat IoT.

Langkah-langkah:

- Buka website Tuya IoT Platform.
- Pilih menu Sign Up.
- Daftarkan akun menggunakan email.
- Verifikasi email.
- Login ke Tuya IoT Platform.

2. Membuat Cloud Project

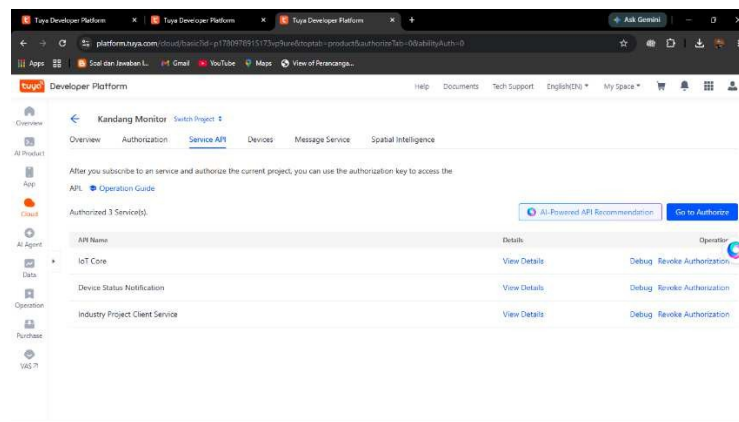
Setelah akun berhasil dibuat, langkah berikutnya adalah membuat Cloud

Project yang berfungsi sebagai wadah pengelolaan API dan perangkat IoT.

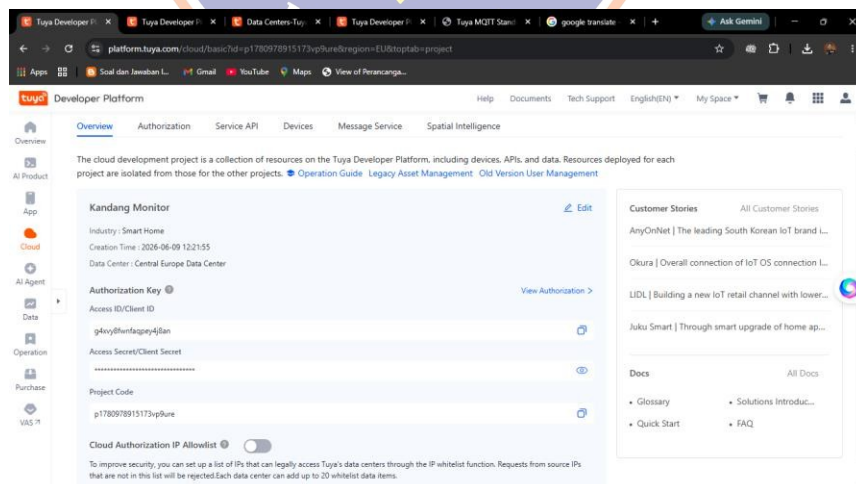
Langkah-langkah:

- Masuk ke menu Cloud.
- Pilih Create Cloud Project.
- Isi informasi proyek:
 - Project Name
 - Industry
 - Development Method
 - Data Center

Aktifkan layanan API yang diperlukan. (Sesuaikan dengan kebutuhan proyek)



Gambar 1.
Menentukan Layanan API



Gambar 2.
Client ID dan Client Secret

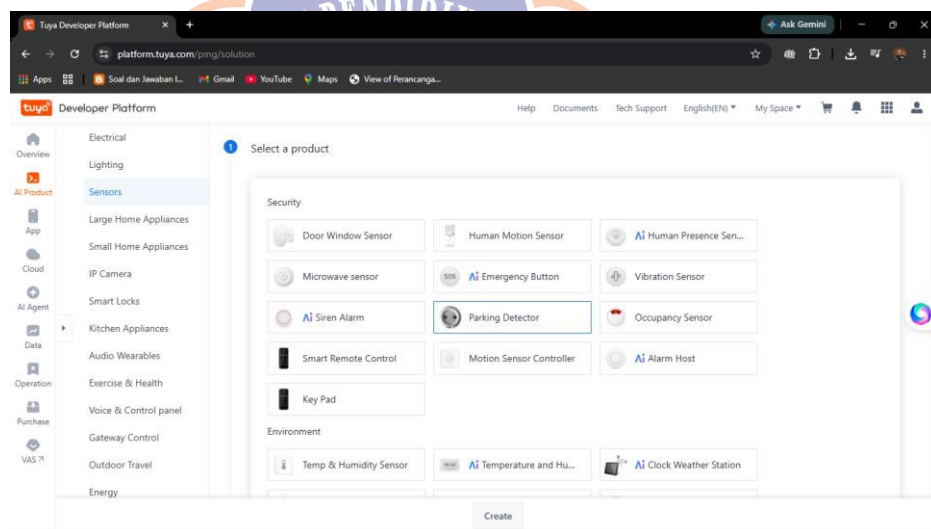
Kedua informasi tersebut digunakan sebagai autentikasi ketika aplikasi melakukan komunikasi dengan Tuya Cloud.

3. Membuat AI Product / Virtual Device

Karena penelitian ini menggunakan dashboard berbasis cloud, maka diperlukan perangkat virtual sebagai media penyimpanan data sensor.

Langkah-langkah:

- Masuk ke menu AI Product.
- Pilih Create Product.
- Tentukan jenis perangkat. (karena saya menggunakan sensor gas, dan ada dht22, saya memilih Temp & Humadity Sensor



Gambar 3.
Menentukan perangkat untuk virtual device

Kemudian setelah pilih Temp & Humadity Sensor Pilih TuyaLink, klik kotak TuyaLink (sebelah kanan) bukan TuyaOS. Kemudian klik Create

Alasannya:

TuyaOS = untuk modul *hardware* Tuya asli

TuyaLink = untuk device pihak ketiga seperti ESP32 kamu yang konek via HTTP API

4. Kemudian akan diminta untuk mengisi beberapa kolom form, isi dengan data seperti dibawah, sesuaikan juga dengan proyek yang dibuat, jika sudah lalu klik Create

Product Name: Kandang Monitor

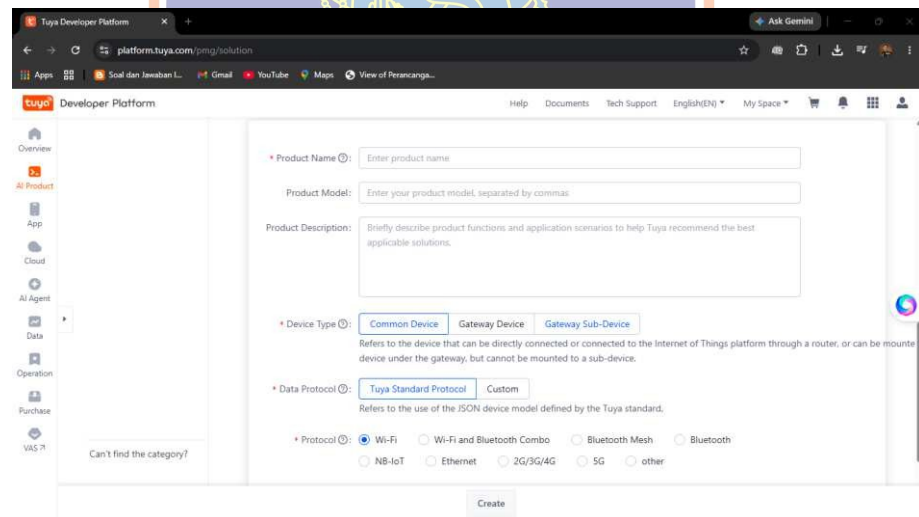
Product Model: ESP32-Node1

Product Description: Monitoring suhu, gas MQ4, dan flow air kandang

Device Type: Common Device

Data Protocol: Custom

Protocol: Wi-Fi



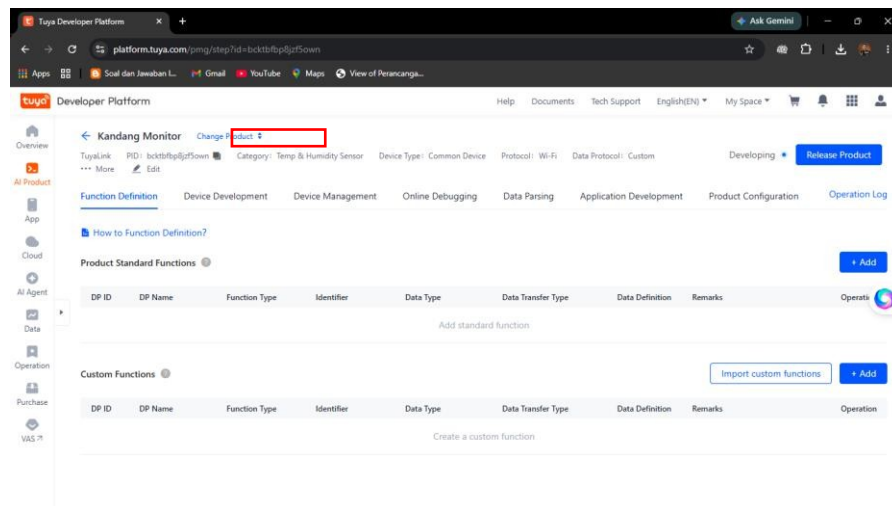
The screenshot shows the Tuya Developer Platform interface. The form is titled 'Create Product' and contains the following fields and options:

- Product Name:** Enter product name
- Product Model:** Enter your product model, separated by commas
- Product Description:** Briefly describe product functions and application scenarios to help Tuya recommend the best applicable solutions.
- Device Type:** Common Device (selected), Gateway Device, Gateway Sub-Device
- Data Protocol:** Tuya Standard Protocol, Custom (selected)
- Protocol:** Wi-Fi (selected), Wi-Fi and Bluetooth Combo, Bluetooth Mesh, Bluetooth, NB-IoT, Ethernet, 2G/3G/4G, 5G, other

A 'Create' button is located at the bottom right of the form.

Gambar 4.
Informasi Produk

Jangan pakai Tuya Standard Protocol karena akan definisikan DP sendiri yang tidak ada di template standar Tuya.



Gambar 5.
Kode PID

5. Menambahkan Custom Functions (DP)

Klik "+ Add" di sebelah kanan bagian Custom Functions (bukan yang atas),
lalu tambahkan DP satu per satu:

DP ID: 1

Function Name: Suhu

Identifier: suhu

Function Type: Property

Data Type: Value

Value Range: Min: -20 ~ Max: 80

Pitch: 1

Scale: 1

Unit: °C

Data Transfer Type: Report Only

DP ID: Enter DP ID
The DP ID is required.

* Function Name: Enter function name

* Identifier: Enter identifier

* Function Type: ☒ Property ☐ Event ☐ Action

* Data Type: Value

* Value Range: Minimum Maximum

* Pitch: Enter pitch

* Scale: 0

Unit: Enter unit

* Data Transfer Type: ☒ Send and Report ☐ Report Only

☐ Send Data

Cancel OK

Gambar 6.
Mengisi Data Custom Functions

DP ID	DP Name	Function Type	Identifier	Data Type	Data Transfer Type	Data Definition	Remarks	Operation
101	Suhu	Property	Suhu	Value	Report Only	Value Range: -20-80 Pitch: 1, Scale: 1, Unit: °C		Edit Delete
102	Gas MQ4 1	Property	gas_mq4_1	Value	Report Only	Value Range: 0-500 Pitch: 1, Scale: 0, Unit: ppm		Edit Delete
103	Gas MQ4 2	Property	gas_mq4_2	Value	Report Only	Value Range: 0-500 Pitch: 1, Scale: 0, Unit: ppm		Edit Delete
104	Status Gas	Property	status_gas	String	Report Only	-		Edit Delete
105	Total Air	Property	total_air	Value	Report Only	Value Range: 0-999999 Pitch: 1, Scale: 0, Unit: mL		Edit Delete

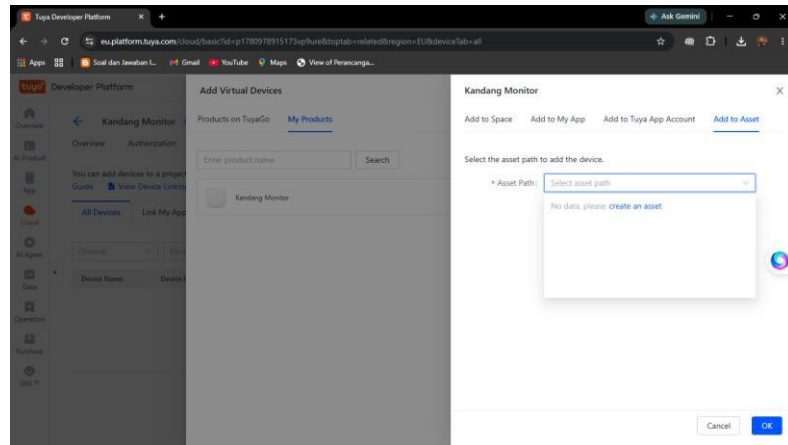
Gambar 7.
Tampilan jika semua DP sudah berhasil dibuat

6. Daftarkan Virtual Device

Sekarang kembali ke Cloud Project untuk menambahkan virtual device menggunakan product ini.

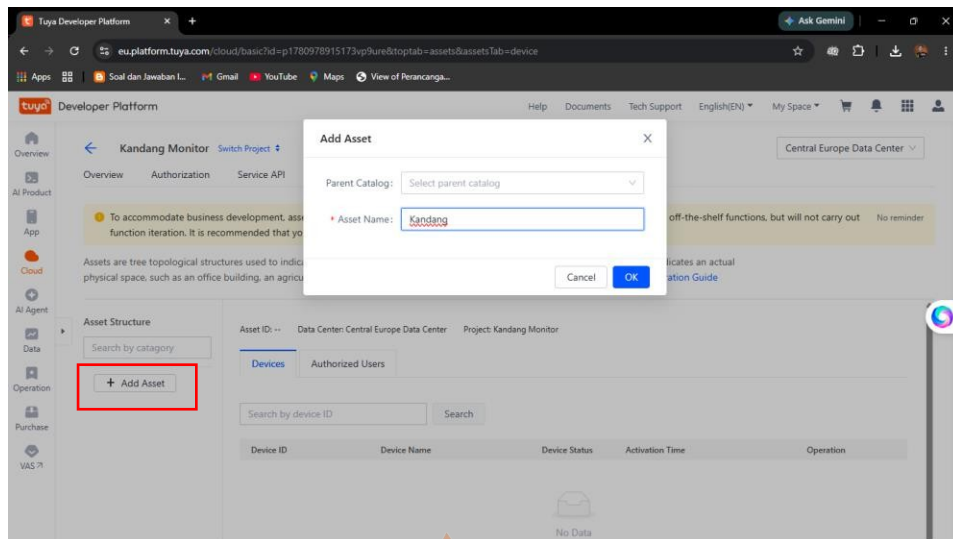
- Klik "Cloud" di sidebar kiri
- Masuk ke project "Kandang Monitor"

- Klik tab "Devices"
- Klik "Add Device" ³ "Add Virtual Device"
- Klik tab "My Products" - product Kandang Monitor akan muncul di sana
- Klik "Add Virtual Devices"
- Klik "Add to Asset" lalu create an asset dan buat nama asset seperti contoh saya menggunakan <Kandang= dan klik simpan



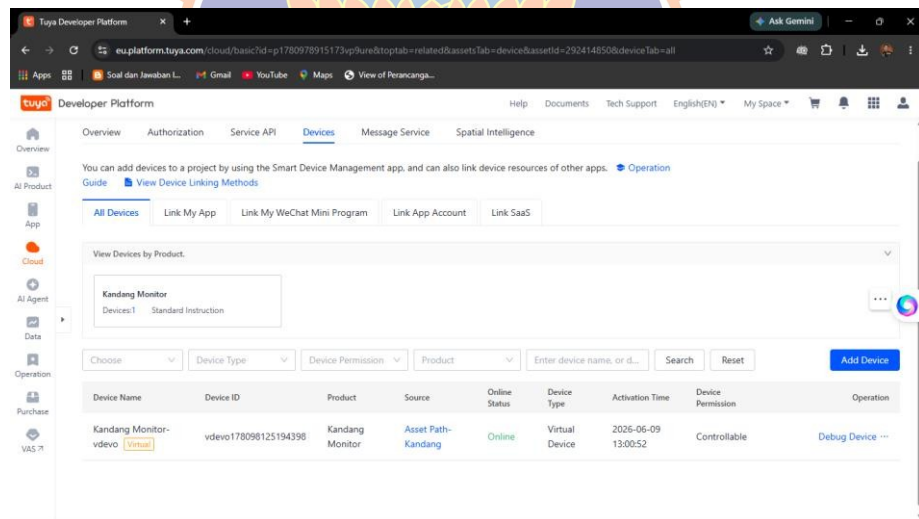
Gambar 8.
Membuat Asset

Kemudian jika sudah membuat dan menambahkan asset yang sudah dibuat, selanjutnya klik <Add Asset= lalu masukan nama asset yang sudah dibuat sebelumnya, lalu klik OK



Gambar 9.
Menambahkan asset yang sudah dibuat

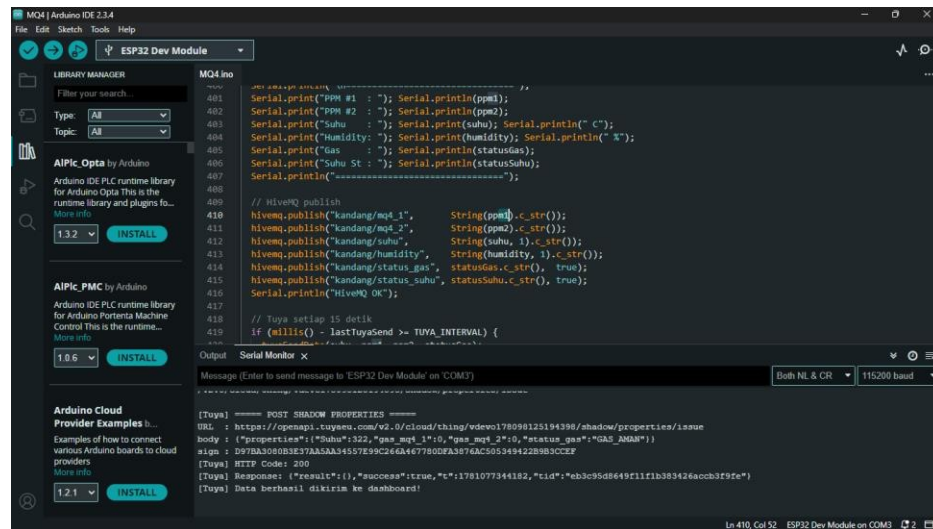
Jika sudah maka akan muncul Device yang sudah dibuat dan statusnya online, lalu akan terlihat Device ID juga yang akan digunakan di program nanti yang di gabungkan dengan Access ID dan Access Secret



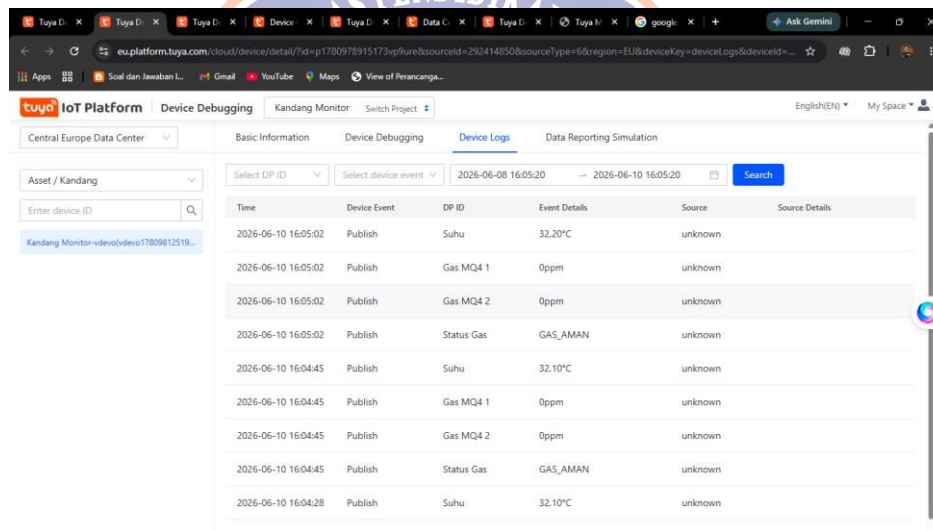
Gambar 10.
Berhasil Membuat Virtual Device

Setelah proses pembuatan Virtual Device selesai, tahap berikutnya adalah melakukan pemrograman pada Arduino IDE untuk mengintegrasikan ESP32 dengan Tuya Cloud API. Pada tahap ini, program dikonfigurasi dengan memasukkan parameter autentikasi yang diperoleh dari Tuya IoT Platform, seperti Access ID, Access Secret, Device ID, serta informasi koneksi lainnya yang diperlukan. Selanjutnya, program diunggah ke ESP32 dan perangkat akan melakukan koneksi ke jaringan Wi-Fi, kemudian menjalankan proses autentikasi ke Tuya Cloud menggunakan kredensial yang telah dikonfigurasi. Apabila proses autentikasi berhasil, Tuya akan mengembalikan Access Token yang digunakan sebagai izin akses untuk mengirimkan data ke Virtual Device.

Setelah Access Token berhasil diperoleh, ESP32 mulai mengirimkan data hasil pembacaan sensor ke Tuya Cloud melalui API sesuai dengan Data Point (DP) yang telah dibuat pada Virtual Device. Keberhasilan proses integrasi dapat diketahui melalui tampilan pada Serial Monitor Arduino IDE, yang menunjukkan pesan bahwa Access Token berhasil diperoleh dan data berhasil dikirim ke Tuya Cloud. Selanjutnya, data tersebut akan ditampilkan secara *real-time* pada dashboard Tuya sehingga pengguna dapat memantau kondisi sistem sesuai dengan hasil pembacaan sensor.



Gambar 11.
Tampilan Serial Monitor

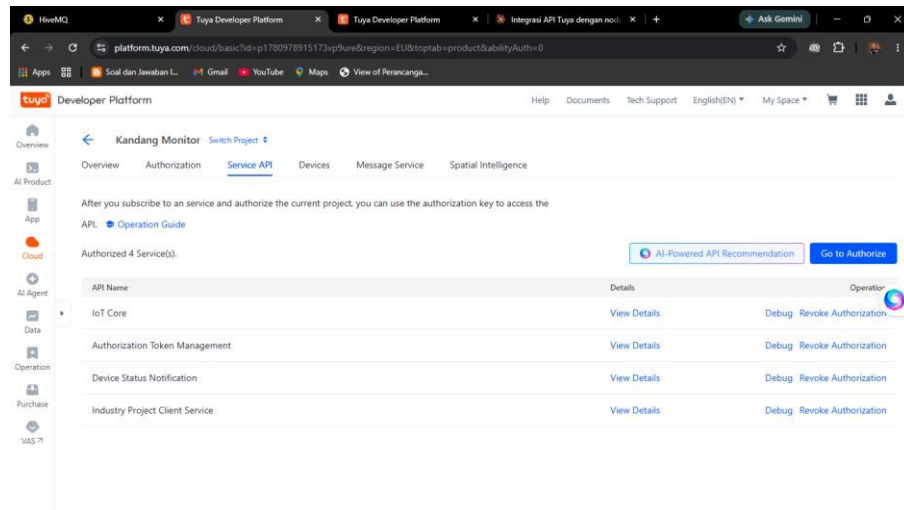


Gambar 12.
Tampilan Dashboard

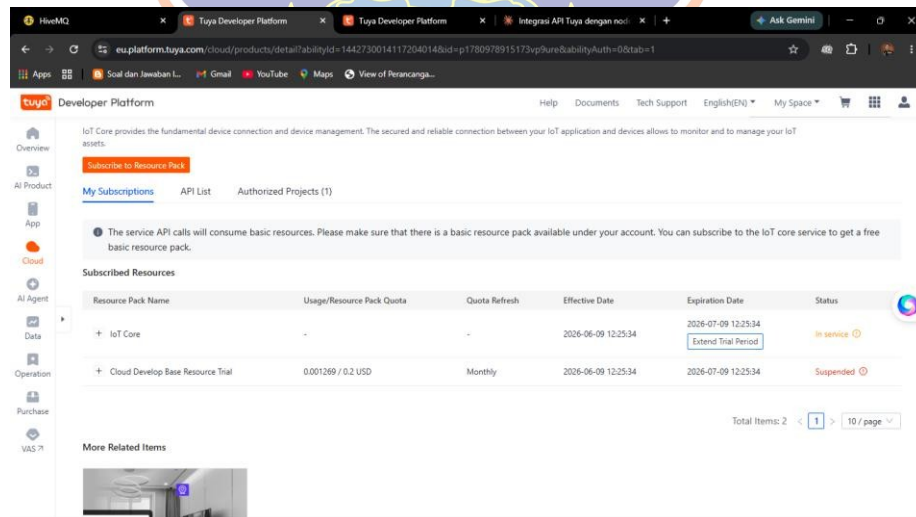
Cara Memperbaiki Service API yang Kadaluarsa.

Tuya memberikan trial gratis untuk layanan IoT Core selama periode tertentu (biasanya 1 bulan). Karena project saya dibuat sekitar bulan Juni 2026, kemungkinan trial sudah habis di Juli ini.

1. Buka eu.platform.tuya.com³ masuk ke project Kandang Monitor
2. Klik tab "Service API"
3. Cari "IoT Core"³ kemungkinan statusnya expired atau perlu diperbarui
4. Klik "Subscribe" atau "Renew" untuk memperbarui layanan IoT Core



Gambar 13.
Service API



Gambar 14.
Extend Trial Period

Disini terlihat pada IoT Core sudah lewat masa berlakunya hingga harus diperpanjang, lalu klik <Extend Trial Period=>

The screenshot shows the 'Extend Service Trial Period' page on the Tuya Developer Platform. The page has a sidebar with navigation links like Overview, AI Product, App, Cloud, AI Agent, Data, Operation, Purchase, and VAS. The main content area contains a form with the following fields:

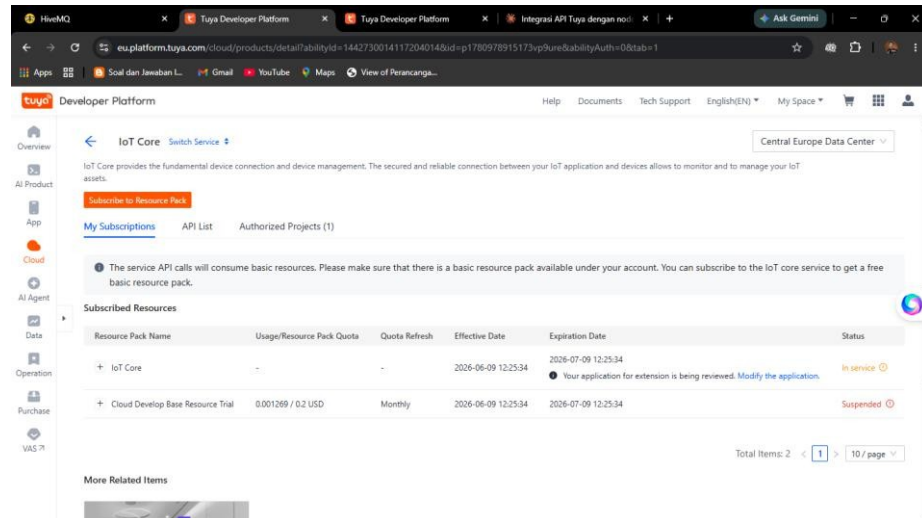
- Extension Period:** A dropdown menu labeled 'Select extension period'.
- Developer Identity:** A dropdown menu labeled 'Select your identity'.
- Estimated Number of Connected Devices:** A dropdown menu labeled 'Select the number of connected devices'.
- Project Overview:** A text area labeled 'Please describe your project background, estimated lead time, and relevant information'.
- Contact Person:** A text input field labeled 'Enter contact person of your project'.
- Contact Information:** A text input field labeled 'Enter phone number or email address'.

At the bottom of the form, there are 'Cancel' and 'Submit' buttons. A note at the top of the form states: 'Please note that extending the trial edition service may result in changes to the related benefits. For more details please refer to pricing document and review the "Free Trial Version" benefits.'

Gambar 15.
Pengisian Data Perpanjang Periode

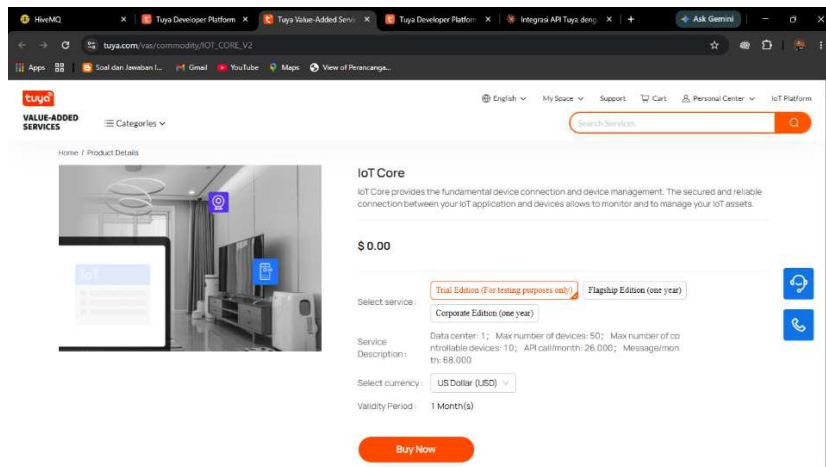
Setelah masa uji coba (trial) layanan Tuya Developer Platform berakhir, dilakukan proses perpanjangan layanan (*Extend Service Trial Period*) agar layanan cloud tetap dapat digunakan. Pada tahap ini, pengguna diminta mengisi beberapa informasi, meliputi Extension Period sebagai lama perpanjangan yang diajukan, Developer Identity sebagai identitas pengembang, Estimated Number of Connected Devices untuk memperkirakan jumlah perangkat yang akan terhubung, Project Overview yang berisi ringkasan proyek, Contact Person sebagai penanggung jawab proyek, serta Contact Information berupa alamat email atau nomor telepon yang dapat dihubungi. Setelah seluruh data diisi dengan lengkap, pengajuan dikirim melalui tombol Submit untuk diproses oleh pihak Tuya. Apabila permohonan disetujui, layanan

cloud akan diperpanjang sehingga proses integrasi dan komunikasi antara perangkat IoT dengan Tuya Cloud dapat terus dilakukan.



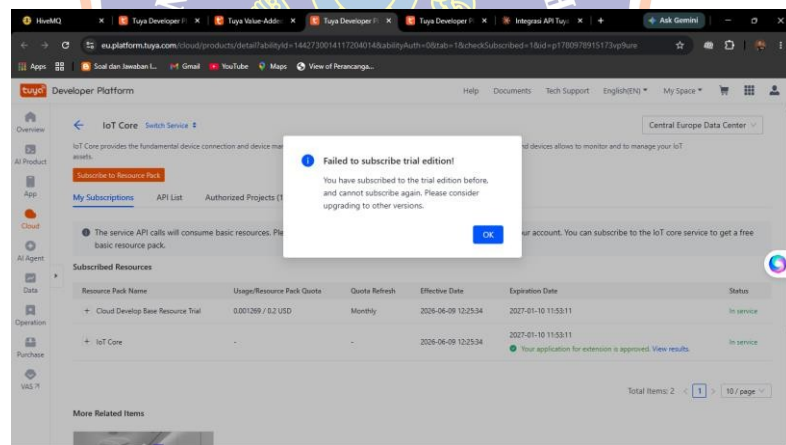
Gambar 16.
Proses Verifikasi

Pada tahap ini terlihat bahwa status Cloud Develop Base Resource Trial masih Suspended, yang menandakan bahwa paket sumber daya (resource pack) belum aktif. Kondisi ini perlu diperbaiki karena resource pack berfungsi menyediakan kuota atau kredit (API call quota) yang diperlukan agar aplikasi dapat berkomunikasi dengan Tuya Cloud melalui API.



Gambar 17.
Pembelian Trial Edition

Jika sudah, kembali ke halaman sebelumnya maka akan terlihat sudah di verifikasi perpanjangan masa aktif dan pembelian berhasil dengan tanda status <in service=



Gambar 18. Resource Kembali aktif

Setelah proses pengajuan perpanjangan Resource Pack berhasil disetujui, status layanan pada Cloud Develop Base Resource Trial berubah menjadi aktif sehingga kuota API call kembali tersedia dan dapat digunakan untuk komunikasi dengan Tuya Cloud. Dengan aktifnya resource pack tersebut, proses autentikasi, pengiriman data

sensor, serta pembaruan data pada *dashboard* Tuya dapat berjalan dengan normal. Berdasarkan informasi pada Tuya Developer Platform, masa berlaku resource pack telah diperpanjang hingga 10 Januari 2027.



RIWAYAT HIDUP



Penulis bernama lengkap Made Wira Prayoga, lahir di Denpasar pada tanggal 27 Juni 2003. Penulis merupakan warga negara Indonesia yang beragama Hindu dan bertempat tinggal di Banjar Peken, Desa Sangsit, Kecamatan Sawan, Kabupaten Buleleng, Bali. Riwayat pendidikan formal penulis diawali di SD Negeri 13

Kesiman, Denpasar, kemudian melanjutkan pendidikan di SD Negeri 1 Banyuasri dan lulus pada tahun 2016. Selanjutnya, penulis menempuh pendidikan di SMP Negeri 2 Singaraja dan lulus pada tahun 2019, kemudian melanjutkan pendidikan di SMK Negeri 3 Singaraja dan lulus pada tahun 2022. Pada tahun 2022, penulis melanjutkan pendidikan tinggi di Universitas Pendidikan Ganesha, Fakultas Teknik dan Kejuruan, Jurusan Teknik Informatika, Program Studi Ilmu Komputer. Saat ini, penulis sedang menyelesaikan tugas akhir berupa skripsi sebagai salah satu syarat untuk memperoleh gelar Sarjana Komputer (S.Kom.) pada Program Studi Ilmu Komputer, Universitas Pendidikan Ganesha.