**Link sourcode [Github NUGRA21](https://github.com/Nugraa21/IOT-TONGSAMPAH)**

**Dahboard** [**Website dahboard**](https://nugraa21.github.io/IOT-TONGSAMPAH/web/index.html)

**Monitorinh Tong sampah berbasih map**

Penulis Ludang prasetyo nugroho ‘225510017’

**Deskripsi projeck**

Projeck ini untuk memudahkan pengangkut sampah agar mudah untuk mengambil sampah di lokasi tertentu sesuwai informasi yang di tampilkan di dashboard dan bisa memonitoring bagian sampah mana yang akan segera penuh dan sudah penuh agar segera di kosong kan .projeck ini menggunakan MQTT sebagai broker untuk mengirimkan sebuah data secara relltime yang akan di tampilkan di dashboard

**Arsitektur Sistem**

Alat ini hanya menggunakan esp32 yang terhubung dengan sensor ultrasonik yang akan mengecek ketinggian smapah di dalam tong sampah dan akan mengirimkan datanya dengan MQTT untuk di tampilkan di dashboard

1. **Sensor**

* **Sensor Ultrasonik**

Sensorini berfungsi untuk mengecek ketinggian sampah dengan mengirimkan sinyal ultrasonic dan akan di pantulkan .dari situlah sensor akan mengidentifikasi ketinggian sampah

1. **Perangkat Mikrokontroler (ESP 32 )**

* **Menerima Data Sensor**

ESP menerima data dari sensor, baik itu data berat maupun tingkat ketinggian.

* **Pengolahan Data**

Data yang diterima kemudian diolah, misalnya untuk menentukan apakah tong sampah sudah penuh atau belum berdasarkan nilai ambang batas tertentu.

* **Transmisi Data**

Data yang telah diolah kemudian dikirimkan ke server MQTT melalui jaringan Wi-Fi.

1. **Jaringan Wi-Fi**

* Konektivitas

Jaringan Wi-Fi berfungsi menghubungkan perangkat ESP dengan server MQTT.

1. **Server MQTT**

* **Broker Pesan**

Server MQTT bertindak sebagai broker pesan yang menghubungkan antara perangkat ESP (sebagai publisher) dengan aplikasi monitoring (sebagai subscriber).

Server MQTT menerima data yang dikirimkan oleh ESP.

* **Penyimpanan Data**

Data yang diterima dapat disimpan dalam database server MQTT untuk keperluan analisis lebih lanjut.

1. **Aplikasi Monitoring**

* **Monitoring**

Untuk memonitoring data yang di terima dari mqtt di sini saya menggunakan website yang terhubung ke mqtt untuk selalu mengambil data bila ada data baru

* **Visualisasi**

Dengan menampilkan map yang akan memunculkan lokasi setiap tong sampah maka oprator sampah dapat melihat dimana likasi sampahnya penuh

* **Notifikasi**

Saat ada sala satu dari tong sampah penuh maka akan memunculkan notifikasi agar segera di ambil dan saat tong sampah sudah di ambil maka notifikasinya akan menghilang

**Deskripsi system**

Sistem monitoring tong sampah ini dirancang untuk memberikan informasi real-time mengenai tingkat kepenuhan tong sampah di berbagai lokasi. Dengan menggunakan teknologi IoT, sistem ini memungkinkan pengumpulan data secara otomatis dan pengiriman data ke platform pusat untuk dianalisis dan divisualisasikan.

**Tujuan Utama**

* **Efisiensi Pengumpulan Sampah**

Memungkinkan petugas kebersihan untuk merencanakan rute pengumpulan sampah secara lebih efektif.

* **Peningkatan Kebersihan Lingkungan**

Mencegah terjadinya penumpukan sampah yang berlebihan.

* **Pengambilan Keputusan yang Lebih Baik**

Memberikan data yang akurat untuk mendukung pengambilan keputusan terkait pengelolaan sampah.

**Fitur Utama**

* **Monitoring Real-time**

Data tingkat kepenuhan tong sampah diperbarui secara berkala.

* **Notifikasi**

Sistem dapat mengirimkan notifikasi ketika tong sampah sudah penuh atau terjadi kesalahan.

* **Visualisasi Data**

Data dapat ditampilkan dalam bentuk Map

* **Integrasi**

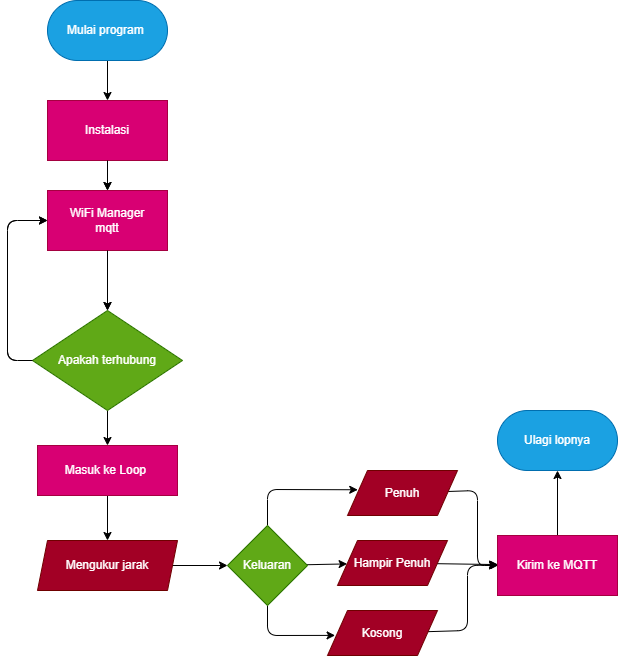
Sistem dapat diintegrasi dengan sistem lain seperti GPS untuk pelacakan kendaraan pengangkut sampah.

**Prinsip Kerja Sistem**

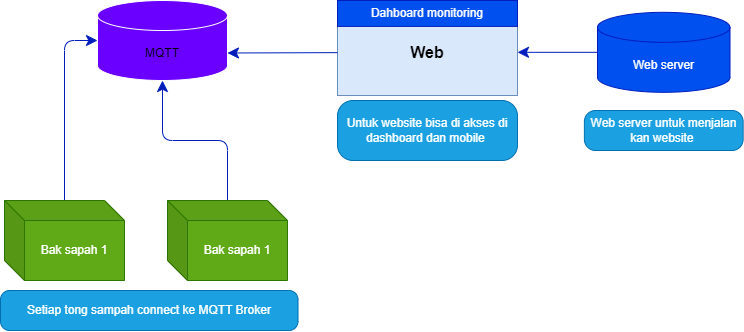
1. **Pengumpulan Data** Sensor pada tong sampah secara terus-menerus mengukur tingkat kepenuhan dan mengirimkan data ke ESP.
2. **Pengolahan Data** ESP menerima data mentah dari sensor, kemudian memprosesnya menjadi data yang lebih bermakna (misalnya, persentase tingkat kepenuhan).
3. **Transmisi Data** Data yang telah diolah dikirimkan ke server MQTT melalui jaringan Wi-Fi.
4. **Penerimaan dan Penyimpanan Data** Server MQTT menerima data dari semua perangkat ESP dan menyimpannya dalam database.
5. **Distribusi Data** Aplikasi monitoring secara berkala mengambil data dari server MQTT dan menampilkannya pada antarmuka pengguna.

**Bentuk arsitektur dan pengetesan**

**Diagram alir**

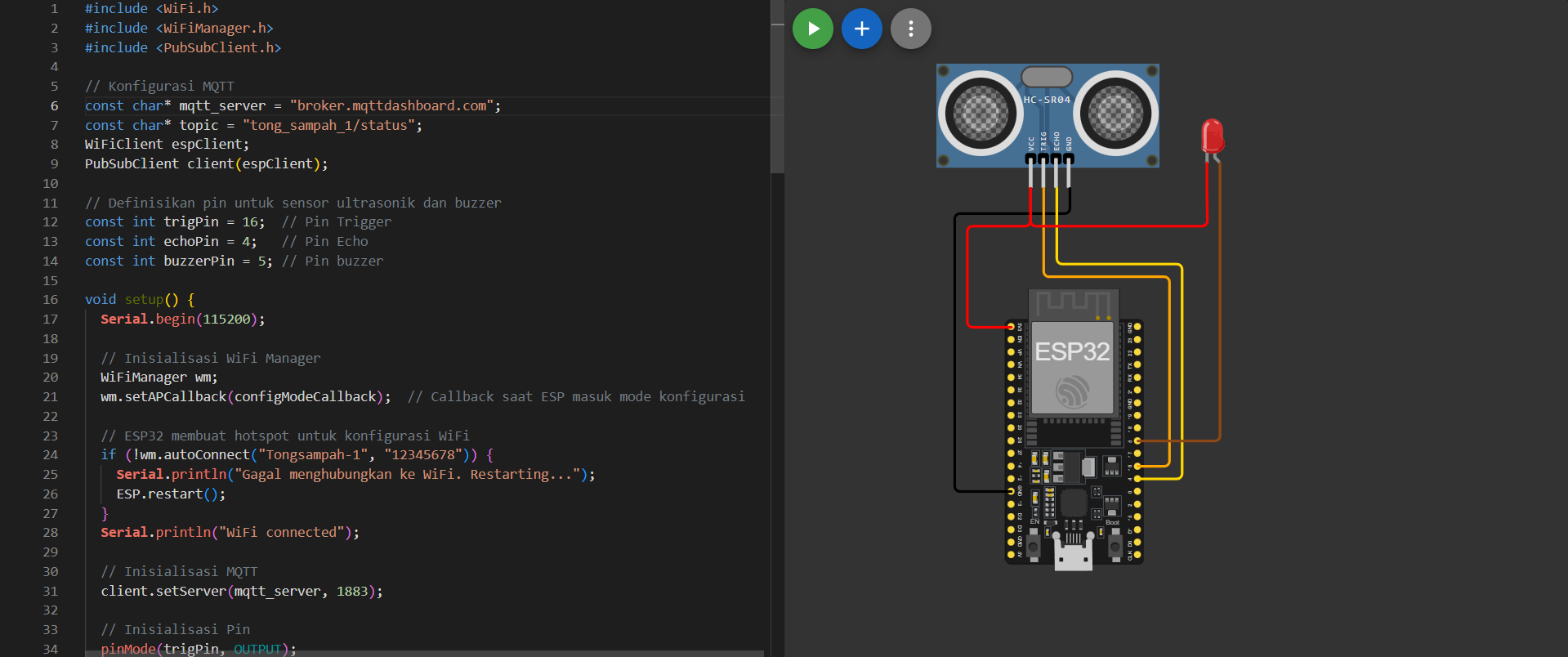


**Diagram block**



**Implementasi hatwere / softwere**

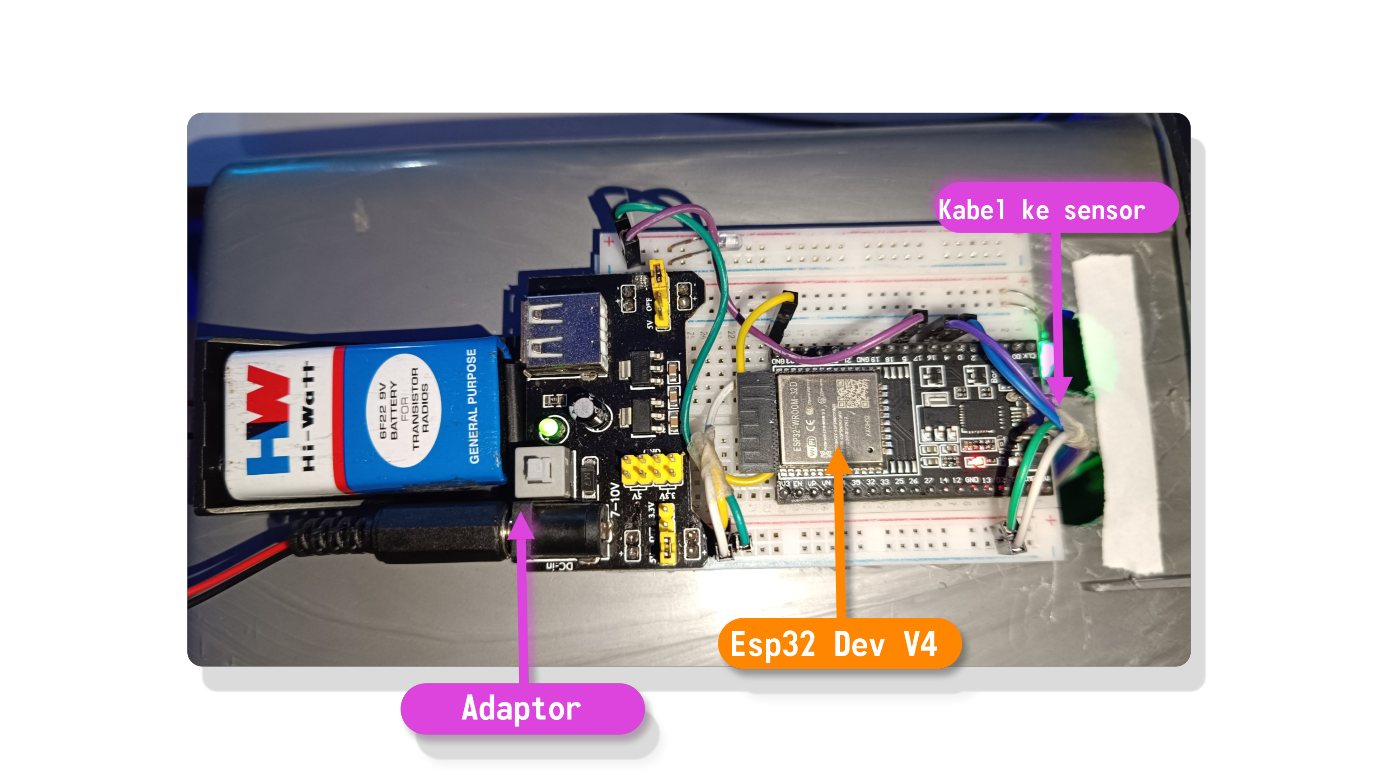
**Implementasi secara simulasi menggunakan wokwi**



* + [Wokwi](https://wokwi.com/projects/417994996499500033) < --- Untuk melihat bentuk secara simulator

Implementasi secara langsung dengan hatwere

Untuk yang pertama ada bagian Modul Esp32 V4 yang berfungsi sebagai transmisi data / mengirimkan data ke mqtt dan menerima data dari sensor



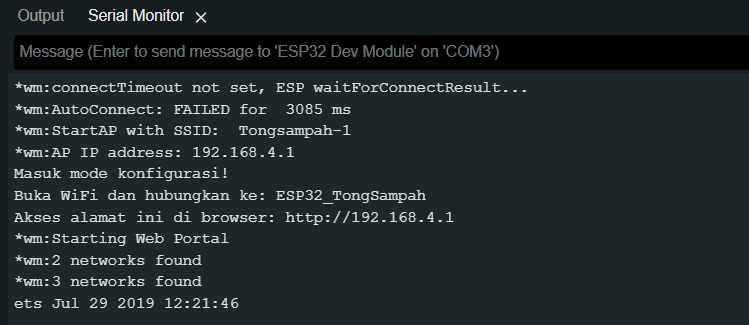
Bagian sensor Ultrasonik berfungsi untuk mendeteksi ketinggian sampah apakah kosong ,hampir penuh , atau sudah penuh



Implementasi program & **Pengetesan code**

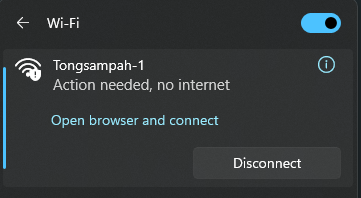
**Code C++ Arduino IDE**

Ini saat program baru saja di uplad maka akan menyalakan wifimanager di esp32 nya dan menyuruh user untuk memasukan ssid dan password yang akan di hubungkan ke esp32 nya agart bisa Online untuk mengirimdata

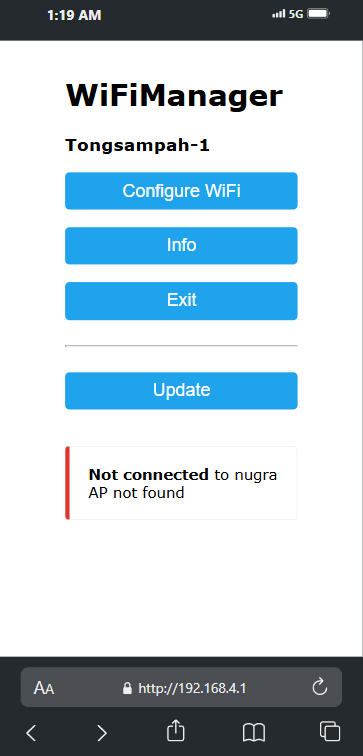
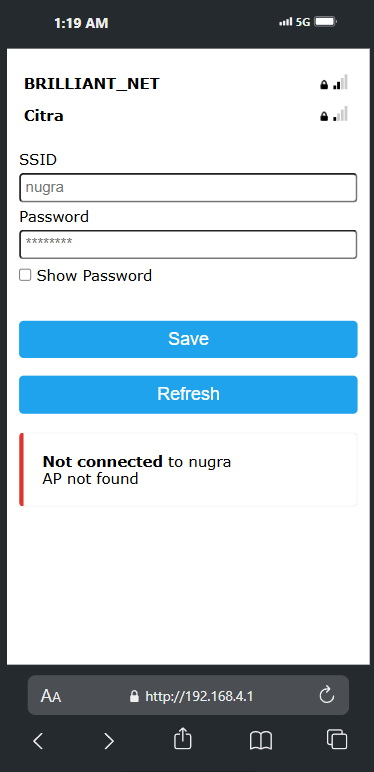




Nama wifi manager dari esp32 kalian harus terhubung ke wifi esp untuk membuka portal untuk wifimanagernya dan mengakses “ http://192.168.4.1/ ” di dalam broser anda

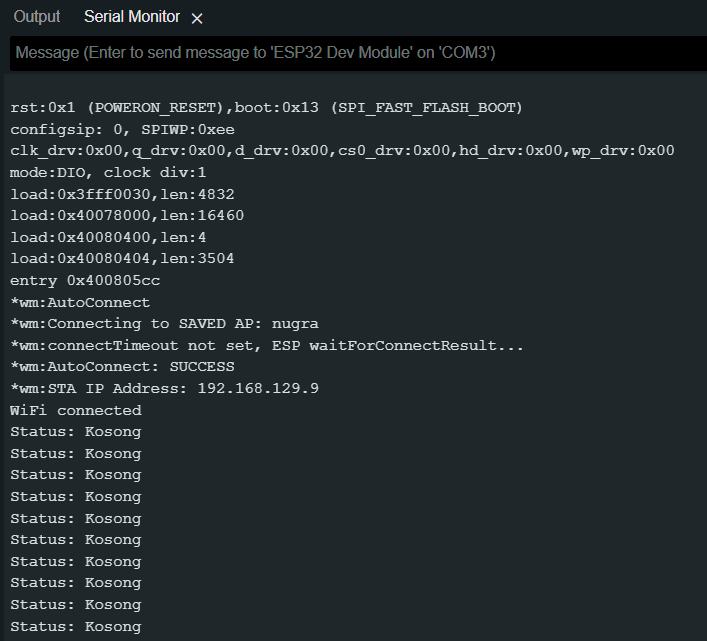


Setelah masuk kalian bisa langsung untuk Configurasi wifinya dan masukan ssid dan password wofo yang akan di sambungkan

Setelah terhubung maka Esp32 akan langsung menyalakan sensor Ultrasoniknya lalu mulai mengirimkan data secara rell tima yang datanya akan di tampilkan di dashboard

“ Dibagian Ip dan ssid AP   
( Akses poin ) add itu adalah wifi yang terhubung ke esp32 ”





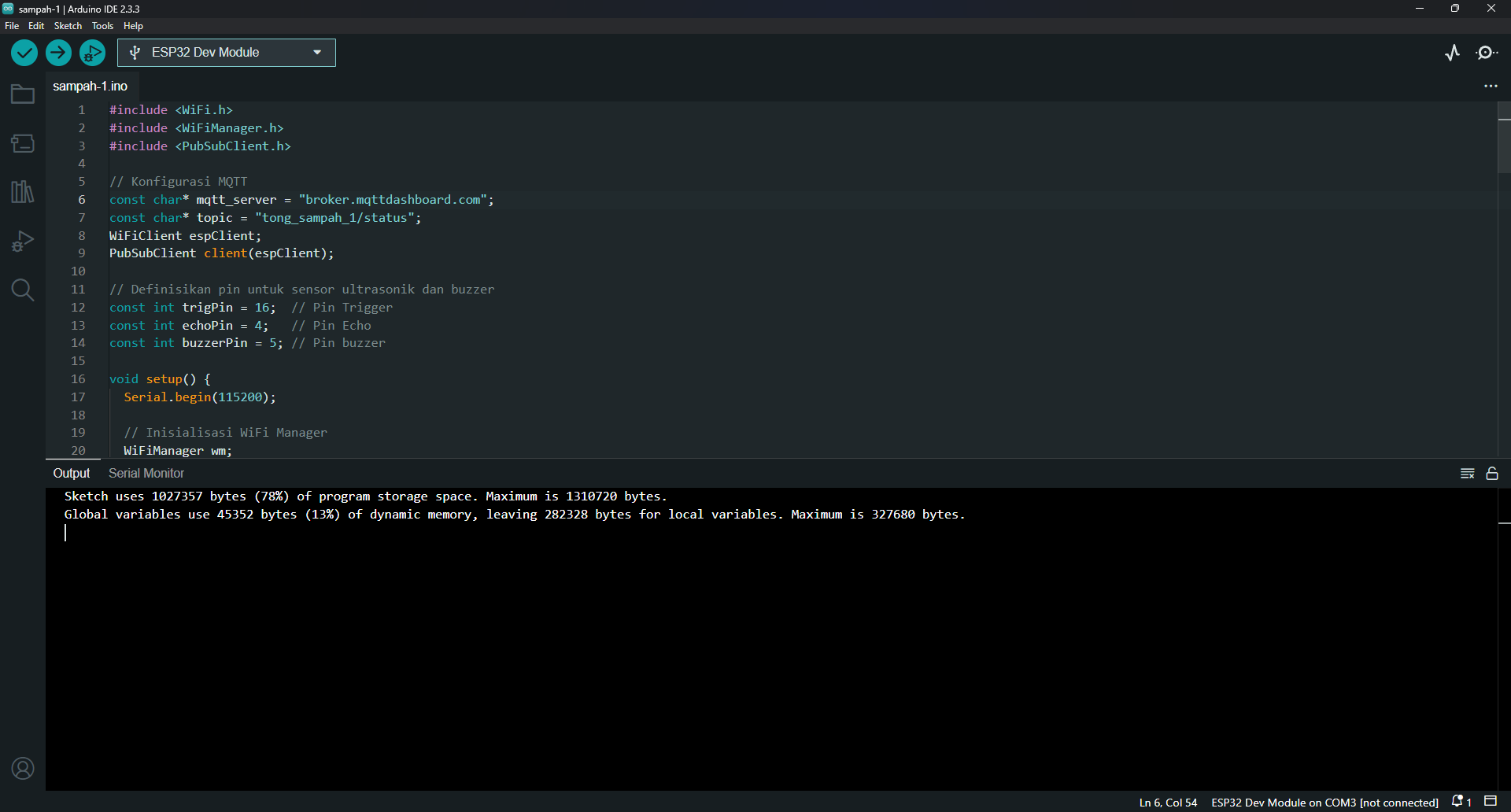
Code program ( Untuk setiap tong sampah akan menyesuwaikan nama nya unruk publiser )

Misal tongsampah1 dan tongsampah2

|  |
| --- |
| #include <WiFi.h> |
| #include <WiFiManager.h> |
| #include <PubSubClient.h> |
|  |
| // Konfigurasi MQTT |
| const char\* mqtt\_server = "broker.mqttdashboard.com"; |
| const char\* topic = "tong\_sampah\_1/status"; |
| WiFiClient espClient; |
| PubSubClient client(espClient); |
|  |
| // Definisikan pin untuk sensor ultrasonik dan buzzer |
| const int trigPin = 16;  // Pin Trigger |
| const int echoPin = 4;   // Pin Echo |
| const int buzzerPin = 5; // Pin buzzer |
|  |
| void setup() { |
| Serial.begin(115200); |
|  |
| // Inisialisasi WiFi Manager |
| WiFiManager wm; |
| wm.setAPCallback(configModeCallback);  // Callback saat ESP masuk mode konfigurasi |
|  |
| // ESP32 membuat hotspot untuk konfigurasi WiFi |
| if (!wm.autoConnect("Tongsampah-1", "12345678")) { |
| Serial.println("Gagal menghubungkan ke WiFi. Restarting..."); |
| ESP.restart(); |
| } |
| Serial.println("WiFi connected"); |
|  |
| // Inisialisasi MQTT |
| client.setServer(mqtt\_server, 1883); |
|  |
| // Inisialisasi Pin |
| pinMode(trigPin, OUTPUT); |
| pinMode(echoPin, INPUT); |
| pinMode(buzzerPin, OUTPUT); |
| } |
|  |
| void loop() { |
| if (!client.connected()) { |
| reconnect(); |
| } |
| client.loop(); |
|  |
| long distance = measureDistance(); |
|  |
| // Tentukan status berdasarkan jarak |
| String status; |
| if (distance > 15) { |
| status = "Kosong"; |
| digitalWrite(buzzerPin, LOW); |
| } else if (distance > 10 && distance <= 20) { |
| status = "Hampir Penuh"; |
| digitalWrite(buzzerPin, LOW); |
| } else if (distance <= 8) { |
| status = "Sudah Penuh"; |
| digitalWrite(buzzerPin, HIGH); |
| } |
|  |
| // Kirim status ke MQTT |
| client.publish(topic, status.c\_str()); |
| Serial.println("Status: " + status); |
|  |
| delay(2000); // Delay untuk pembacaan berikutnya |
| } |
|  |
| // Fungsi untuk mengukur jarak |
| long measureDistance() { |
| digitalWrite(trigPin, LOW); |
| delayMicroseconds(2); |
| digitalWrite(trigPin, HIGH); |
| delayMicroseconds(10); |
| digitalWrite(trigPin, LOW); |
|  |
| long duration = pulseIn(echoPin, HIGH); |
| long distance = duration \* 0.034 / 2; |
| return distance; |
| } |
|  |
| // Fungsi untuk reconnect MQTT |
| void reconnect() { |
| while (!client.connected()) { |
| if (client.connect("ESP32\_TongSampah\_1")) { |
| client.subscribe(topic); |
| } else { |
| delay(5000); |
| } |
| } |
| } |
| // Callback saat ESP32 masuk mode konfigurasi |
| void configModeCallback(WiFiManager \*myWiFiManager) { |
| Serial.println("Masuk mode konfigurasi!"); |
| Serial.println("Buka WiFi dan hubungkan ke: ESP32\_TongSampah"); |
| Serial.println("Akses alamat ini di browser: http://192.168.4.1"); |
| } |

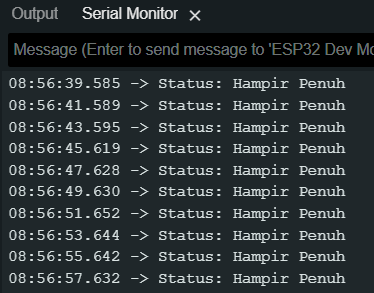
**Pengetesan code**

Code berjalan sukses dan tanpa kesalahan siap untuk di gunakan

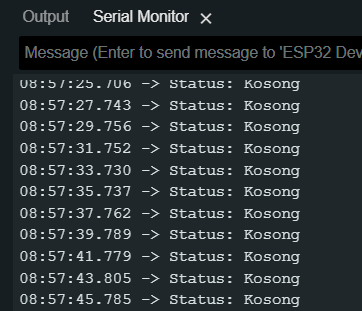
****

Pengetesan keluaran program apakah tidak ada masalah

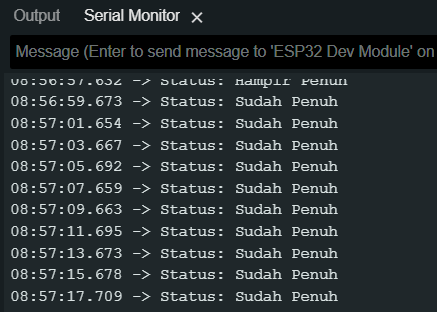
Keluaran saat di bawh 10 cm akan mengirimkan data hampir penuh

****

**Keluaran saat di di bawah 8 cm maka akn mengirimkan data bahwa sudah penuh**



**Dan saat di atas 10 maka akan megirimkan data bahwa kosong**



**Code Js { Yang sudah di sederhanakan }**

**Code untuk instalasi map bagian const map itu adalah titik kordinat awal saat kita memasukan map jadi saat pertama kali masuk ke map maka akan menuju ke koordinat tersebut**

*// Inisialisasi peta*

*const* map = L.map('map').setView([-7.8204, 110.4266], 20); *// Pusat peta*

L.tileLayer('https://{s}.tile.openstreetmap.org/{z}/{x}/{y}.png', {

  attribution: '&copy; <a href="https://www.openstreetmap.org/copyright">OpenStreetMap</a> contributors'

}).addTo(map);

*// Fungsi untuk membuat marker dan pop-up*

*function* createMarker(*tongSampah*) {

*const* marker = L.marker(tongSampah.lokasi, {

      icon: L.divIcon({ className: 'marker-empty', html: '🟢' })

  }).addTo(map)

      .bindPopup(`

          <div style="font-family: Arial, sans-serif; padding: 10px;">

              <h3 style="margin: 0; font-size: 18px;">${tongSampah.nama}</h3>

              <p style="margin: 5px 0;">Alamat: Sekarsuli, Sendangtirto, Kec. Berbah, Kabupaten Sleman, Daerah Istimewa</p>

              <p id="statusTongSampah${tongSampah.id}" class="status-empty-js" style="margin: 5px 0; font-weight: bold; color: green;">Status: Kosong</p>

          </div>

      `);

*// Mengembalikan marker untuk referensi di masa depan*

  return marker;

}

**Ini adalah code untuk menyimpan data / variable setiap tong sampah seperti nama ,subacribe , dan koordinat lokasi**

*// Lokasi tong sampah*

*const* lokasiTongSampah = [

  { id: 1, lokasi: [-7.82062992849643, 110.42654669574745], nama: 'Rumah Ludang' },

  { id: 2, lokasi: [-7.820135382837104, 110.42666630956253], nama: 'Tong Sampah 2' },

  { id: 3, lokasi: [-7.821398025909336, 110.42598761336608], nama: 'Tong Sampah 3' }

];

client.on('connect', *function* () {

  console.log('MQTT Connected');

*// Subscribe untuk mendapatkan data dari broker*

  client.subscribe('tong\_sampah\_1/status');

  client.subscribe('tong\_sampah\_2/status');

  client.subscribe('tong\_sampah\_3/status');

});

**Code untuk menambil / mengoneksikan ke MQTT broker**

*const* client = mqtt.connect('ws://broker.mqttdashboard.com:8000/mqtt');

**Code Html { Untuk menampilkan map atau berberapa pop up }**

Code untuk menampilkan map di website

<div *id*="map" *class*="map-container"></div>

**Menampilkan Cart untuk rumah/ tongsampah**

      <div *class*="cart">

        <div *class*="cart-data">

          <div *class*="cart-row-logo">

            <h1><img *width*="30px" *src*="https://cdn-icons-png.flaticon.com/128/18397/18397677.png" *alt*=""> Rumah 1</h1>

            <div *class*="buttons">

              <button *id*="light-theme" *onclick*="location.href='https*://maps.app.goo.gl/g2iWwM3xYWAd1wgC7'*"><img *width*="15px" *src*="https://cdn-icons-png.flaticon.com/128/684/684908.png" *alt*="">Menuju lokasi</button>

          </div>

          </div>

          <div *class*="cart-alamat">

*<!-- <button onclick="showTongSampah(1)">Lihat Lokasi Tong Sampah 1</button> -->*

            <h4><img *width*="15px" *src*="https://cdn-icons-png.flaticon.com/128/1077/1077012.png" *alt*="">    - Rumah milik  : . . . . . . </h4>

            <h4>alamat rumah <a *class*="map-lokasi" *href*="javascript:void(0);" *id*="showLocationButton1">Lihat lokasi</a>

              <p>Sekarsuli, Sendangtirto, Kec. Berbah, <br> Kabupaten Sleman, Daerah Istimewa</p>

            </h4>

          </div>

        </div>

        <div *class*="cart-status">

          <div *id*="statusTongSampah1" *class*="status">Menunggu status...</div>

        </div>

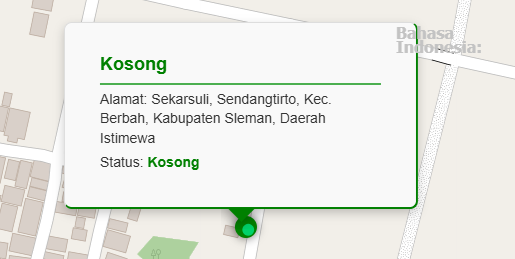
      </div>

**Dahboard & Pengetesan keluaran data**

**Pengetesan keluaran data**

**Saat Tong sampah kosong / sensor tidak mendeteksi si atas 10 cm maka akan mengirimkan data bbila kosong**

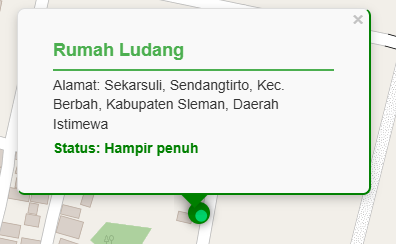
# Ini bentuk pop up daat kosond

****

# Di sini ada bentuk cart yang menunjukan data dari rumah 1 seperti lokasi alamat dll

****

# Bila sensor mendeteksi di bawah 10 cm maka akan memunculkan bahwa hampir penuh



# Di sini ada bentuk cart yang menunjukan data dari rumah 1 seperti lokasi alamat dll

****

# Saat sensor mendeteksi di atas 8 cm maka akan mengirimkan data bahwa sudah penuh dan segera di kosongkan

****

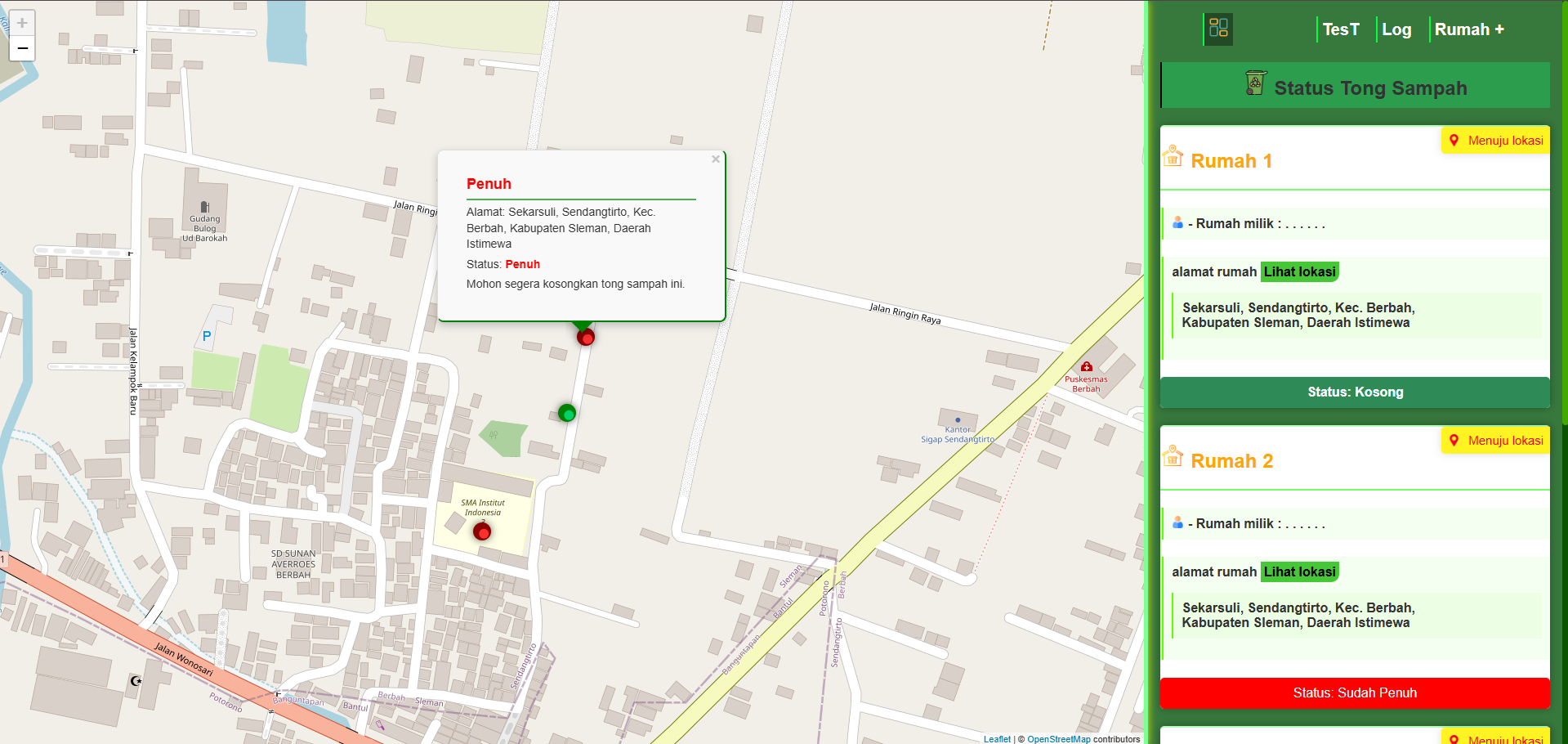
# Di sini ada bentuk cart yang menunjukan data dari rumah 1 seperti lokasi alamat dll

****

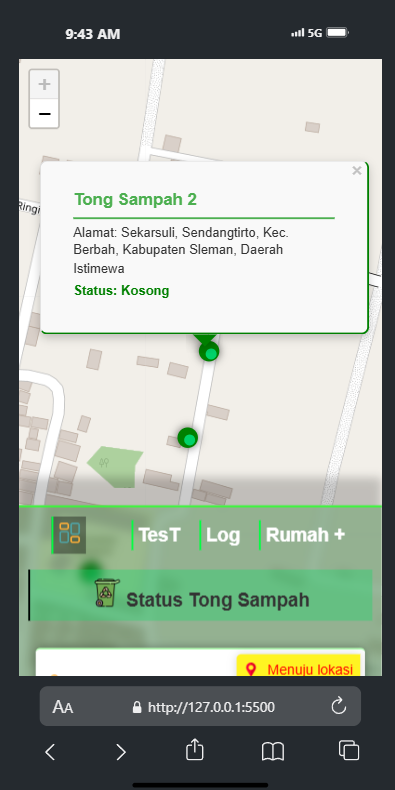
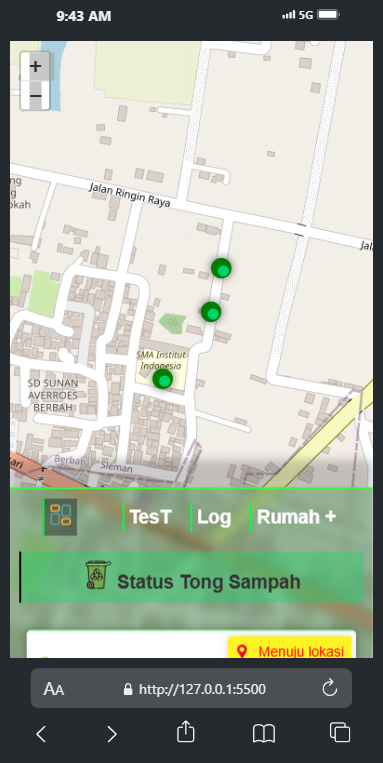
**Dahboard**

**Bentuk dashboard dat ada di tampilan dekstop menggunakan Html css dan js**

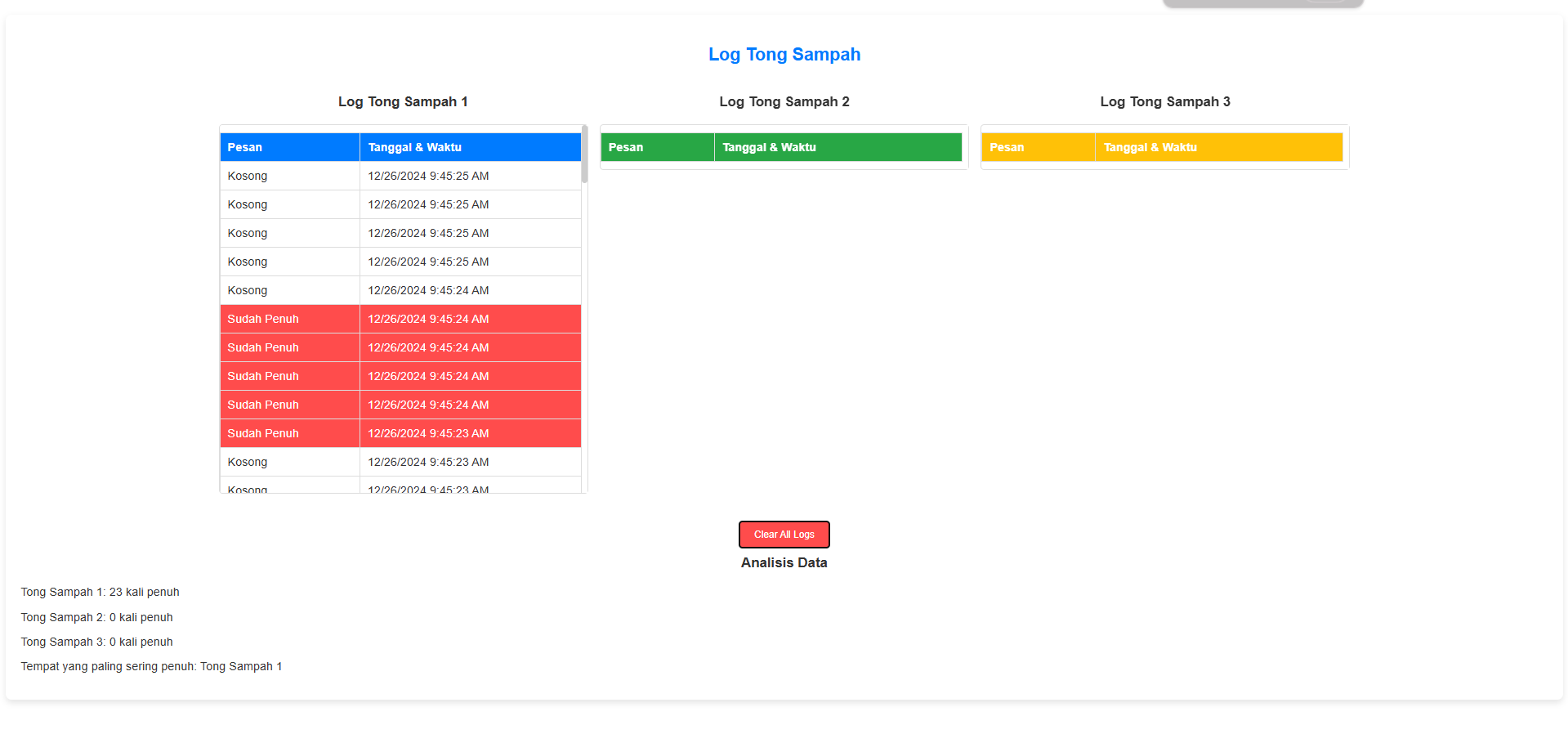
**Contoh saat ada tong sampah yang penuh maka akan memunculkan pop up dan notifikasi**

****

**Ini bentuk reponsif saat di buka ti tampilan mobile**

****

**Aku juga membuat log untuk setiap keluaran di setiap tong sampah dan akan memunculkan log secara rell tame dan aan ada analisis data tong sampah mana yang akan sering penuh**

****

**Thnks : )**