

LAMPIRAN

Source Code Arduino IDE

```

#define BLYNK_TEMPLATE_ID "TMPL69-nEjrTI"
#define BLYNK_TEMPLATE_NAME "Quickstart Template"
#define BLYNK_AUTH_TOKEN "szOeRsobiQ5OnKVVJP56LIqJnggIFi7B"

#define BLYNK_PRINT Serial

#include <ESP8266WiFi.h>
#include
<BlynkSimpleEsp8266.h>
#include <Servo.h>
#include <DHT.h>
#include <SoftwareSerial.h>
#include <TinyGPSPlus.h>
#include <FirebaseESP8266.h>

// === WiFi & Blynk ===
char ssid[] = "conny";
char pass[] =
"444555999";

// === Firebase ===
#define FIREBASE_HOST "smart-waste-management-7c9d5-default-
rtdb.firebaseio.com"
#define FIREBASE_AUTH "rqeXvItDJrwnCjb5fqkYltBxhBRAVCiU2H157p0C"
FirebaseData firebaseData;
FirebaseJson gpsJson;
FirebaseJson logJson;

// === Sensor & Servo===
#define DHTPIN D5
#define DHTTYPE DHT11
DHT dht(DHTPIN, DHTTYPE);

const int trigPin =
D6; const int echoPin
= D7;
Servo servo1; // untuk buka-tutup
Servo servo2; // untuk sortir

long duration;
int distance;
int Ultra_Distance = 15;

```

```
int maxDryHumidity = 60;

// === GPS ===
SoftwareSerial gpsSerial(D2, D1); // RX,
TX TinyGPSPlus gps;

BlynkTimer timer;

void kirimGPS() {
  if (gps.location.isValid()) {
    float lat = gps.location.lat();
    float lng = gps.location.lng();

    Serial.print("GPS: ");
    Serial.print(lat, 6);
    Serial.print(", ");
    Serial.println(lng, 6);

    Blynk.virtualWrite(V6, lat);
    Blynk.virtualWrite(V5, lng);

    gpsJson.clear();
    gpsJson.set("latitude", lat);
    gpsJson.set("longitude", lng);
    gpsJson.set("timestamp", millis() / 1000);

    if (Firebase.setJSON(firebaseData,
"sampah/tempat1/gps", gpsJson)) {
      Serial.println("GPS dikirim ke Firebase.");
    } else {
      Serial.print("Gagal kirim GPS: ");
      Serial.println(firebaseData.errorReason());
    }
  } else {
    Serial.println("GPS belum terkunci.");
  }
}

void sendSensorData() {
  // Baca jarak ultrasonik
  digitalWrite(trigPin, LOW);
  delayMicroseconds(5);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);
```

```

duration = pulseIn(echoPin, HIGH);
distance = duration * 0.034 / 2;

float humidity = dht.readHumidity();
String statusSampah = "Tidak Dikenali";

if (distance < Ultra_Distance && distance > 0) {
  Blynk.virtualWrite(V4, "Ada Sampah");
  Serial.println("Sampah Terdeteksi!");

  servo1.write(180); // buka
  delay(2000);

  if (!isnan(humidity)) {
    Serial.print("Kelembaban: ");
    Serial.print(humidity);
    Blynk.virtualWrite(V2, humidity);

    if (humidity > maxDryHumidity) {
      Serial.println(" --> Organik");
      servo2.write(170);
      Blynk.virtualWrite(V3, "Organik");
      statusSampah = "Organik";
    } else {
      Serial.println(" --> Anorganik");
      servo2.write(10);
      Blynk.virtualWrite(V3, "Anorganik");
      statusSampah = "Anorganik";
    }
  } else {
    Serial.println("Sensor DHT gagal.");
  }

  delay(1000);
  servo2.write(90);
  servo1.write(0); // tutup

  // Kirim ke Firebase
  logJson.clear();
  logJson.set("kelembaban",
humidity);
  logJson.set("status_sampah", statusSampah);
  logJson.set("jarak_cm", distance);
  logJson.set("timestamp", millis() / 1000);

  if (Firebase.pushJSON(firebaseData, "/sampah/logs", logJson)) {

```

```
        Serial.println("Data SAMPAH berhasil dikirim ke Firebase.");
    } else {
        Serial.print("Gagal kirim data: ");
        Serial.println(firebaseData.errorReason());
    }
} else {
    Blynk.virtualWrite(V4, "Kosong");
    Serial.println("Tidak ada sampah.");
}
}

void setup() {
    Serial.begin(115200);
    gpsSerial.begin(9600);

    Blynk.begin(BLYNK_AUTH_TOKEN, ssid, pass);
    Firebase.begin(FIREBASE_HOST, FIREBASE_AUTH);
    Firebase.reconnectWiFi(true);

    pinMode(trigPin, OUTPUT);
    pinMode(echoPin, INPUT);

    dht.begin();
    servo1.attach(D4);
    servo2.attach(D3);
    servo1.write(0);
    servo2.write(90);

    timer.setInterval(5000L, kirimGPS);
    timer.setInterval(5000L,
    sendSensorData);
}

void loop() {
    Blynk.run();
    timer.run();

    while (gpsSerial.available()) {
        gps.encode(gpsSerial.read());
    }
}
```

Source Code Android Studio

1. MapsActivity.kt

```

package com.example.smarttrashmonitor

import android.Manifest
import android.content.pm.PackageManager
import android.os.Bundle
import android.util.Log

import android.app.NotificationChannel
import android.app.NotificationManager
import android.content.Context
import android.os.Build

import androidx.appcompat.app.AppCompatActivity
import androidx.core.app.ActivityCompat
import androidx.core.app.NotificationCompat

import androidx.core.app.NotificationManagerCompat
import com.google.android.gms.maps.CameraUpdateFactory
import com.google.android.gms.maps.GoogleMap
import com.google.android.gms.maps.OnMapReadyCallback
import com.google.android.gms.maps.SupportMapFragment
import com.google.android.gms.maps.model.LatLng
import com.google.android.gms.maps.model.MarkerOptions
import com.google.firebase.database.*

class MapsActivity : AppCompatActivity(), OnMapReadyCallback {

    private lateinit var mMap: GoogleMap
    private lateinit var database:
    DatabaseReference private val channelId
    = "smart_trash_channel"

    override fun
    onCreate(savedInstanceState: Bundle?)
    { super.onCreate(savedInstanceState)
    setContentView(R.layout.activity_maps)

```

```
        val mapFragment =
supportFragmentManager.findFragmentById(R.id.map_fragment)
as SupportMapFragment
        mapFragment.getMapAsync(this)

        database =
        FirebaseDatabase.getInstance().reference.child("sampah")

        createNotificationChannel()
    }

    override fun onMapReady(googleMap:
        GoogleMap) { mMap = googleMap

        if
        (ActivityCompat.checkSelfPermission(this,
        Manifest.permission.ACCESS_FINE_LOCATION)
        == PackageManager.PERMISSION_GRANTED) {
            mMap.isMyLocationEnabled = true
        } else {
            ActivityCompat.requestPermiss
            ions( this,
            arrayOf(Manifest.permission.ACCESS_FINE_LOCATION
            ), 1001
            )
        }
    }

    loadTrashLocations()
}

private fun loadTrashLocations() {
    database.addListenerForSingleValueEvent(object :
    ValueEventListener {
        override fun onDataChange(snapshot: DataSnapshot)
        { mMap.clear()
            for (trashSnapshot in snapshot.children) {
```

```

        val lat =
            trashSnapshot.child("gps/latitude").getValue(Double::class.java)
        val lng =
            trashSnapshot.child("gps/longitude").getValue(Double::class.java)
        val time =
            trashSnapshot.child("gps/timestamp").getValue(Long::class.java)
        ?.toString() ?: "Unknown"

        if (lat != null && lng != null) {
            val location = LatLng(lat, lng)
            mMap.addMarker(
                MarkerOptions()
                    .position(location)
                    .title("Tempat Sampah")
                    .snippet("Update: $time")
            )
            mMap.moveCamera(CameraUpdateFactory.newLatLngZoom(location,
17f
            ))

            showNotification("Tempat Sampah Terupdate", "Update pada: $time")
        }
    }

    override fun onCancelled(error: DatabaseError) {
        Log.e("Firebase", "Error: ${error.message}")
    }
}

private fun showNotification(title: String, message: String) {
    val builder = NotificationCompat.Builder(this, channelId)
        .setSmallIcon(R.drawable.ic_launcher_foreground)
        .setContentTitle(title)
        .setContentText(message)
        .setPriority(NotificationCompat.PRIORITY_HIGH)
}

```

```

        with(NotificationManagerCompat.from(this)) {
            notify(System.currentTimeMillis().toInt(), builder.build())
        }
    }

    private fun createNotificationChannel() {
        if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.O) {
            val channel = NotificationChannel(channelId, "Smart Trash
Notification", NotificationManager.IMPORTANCE_HIGH)
            val manager =
getSystemService(Context.NOTIFICATION_SERVICE) as
NotificationManager
            manager.createNotificationChannel(channel)
        }
    }
}

```

2. AndroidManifest.xml

```

<manifest
    xmlns:android="http://schemas.android.com/apk/res/a
ndroid"
    xmlns:tools="http://schemas.android.com/tools" *
    package="com.example.smarttrashmonitor">

    <uses-permission android:name="android.permission.INTERNET" />

    <uses-permission
        android:name="android.permission.ACCESS_FINE_LOCATION"
    />

    <uses-permission
        android:name="android.permission.ACCESS_COARSE_LOCATION" />

    <uses-permission
        android:name="android.permission.ACCESS_BACKGROUND_LOCATION" />
    <uses-permission
        android:name="android.permission.POST_NOTIFICATIONS"
        tools:targetApi="33" />

    <application
        android:allowBackup="true"

```

```
android:label="Smart Trash Monitor"
android:icon="@mipmap/ic_launcher"
android:roundIcon="@mipmap/ic_launcher_round"
android:theme="@style/Theme.SmartTrashMonitor">

<meta-data
    android:name="com.google.android.geo.API_KEY"
    android:value="AIzaSyCJzOYLqH4KB3zpqwTuhy3ahfUrwhx9AVg" />

<activity
    android:name=".MapsActivity"
    android:exported="true">
    <intent-filter>
        <action android:name="android.intent.action.MAIN" />
        <category android:name="android.intent.category.LAUNCHER" />
    </intent-filter>
</activity>

<service
    android:name=".MyFirebaseMessagingService" android:exported="false">
    <intent-filter>
        <action android:name="com.google.firebase.MESSAGING_EVENT" />
    </intent-filter>
</service>
</application>
</manifest>
```

3. Buildgradle.kts

```
plugins {
    id("com.android.application")
    id("org.jetbrains.kotlin.android")
    id("com.google.gms.google-services")
}

android {
    namespace =
    "com.example.smarttrashmonitor"
    compileSdk = 34

    defaultConfig {
        applicationId =
        "com.example.smarttrashmonitor" minSdk
        = 24
        targetSdk = 34
        versionCode = 1
        versionName = "1.0"

        testInstrumentationRunner =
        "androidx.test.runner.AndroidJUnitRunner"
    }

    buildTypes {
        release {
            isMinifyEnabled = false
            proguardFiles(
                getDefaultProguardFile("proguard-android-
                optimize.txt"), "proguard-rules.pro"
            )
        }
    }

    compileOptions {
        sourceCompatibility = JavaVersion.VERSION_1_8
    }
}
```

```
        targetCompatibility = JavaVersion.VERSION_1_8
    }

    kotlinOptions {
        jvmTarget =
            "1.8"
    }

    buildFeatures {
        viewBinding = true
    }
}

dependencies {
    implementation(platform("com.google.firebase:firebase-
        bom:33.15.0"))
    implementation("com.google.firebase:firebase-
        database-ktx")
    implementation("com.google.firebase:firebase-
        analytics")
    implementation("com.google.firebase:firebase-
        messaging")

    implementation("androidx.core:core-ktx:1.12.0")
    implementation("androidx.appcompat:appcompat:1.6.1")
    implementation("com.google.android.material:material:1.11.0")
    implementation("androidx.constraintlayout:constraintlayout:2.1.4")

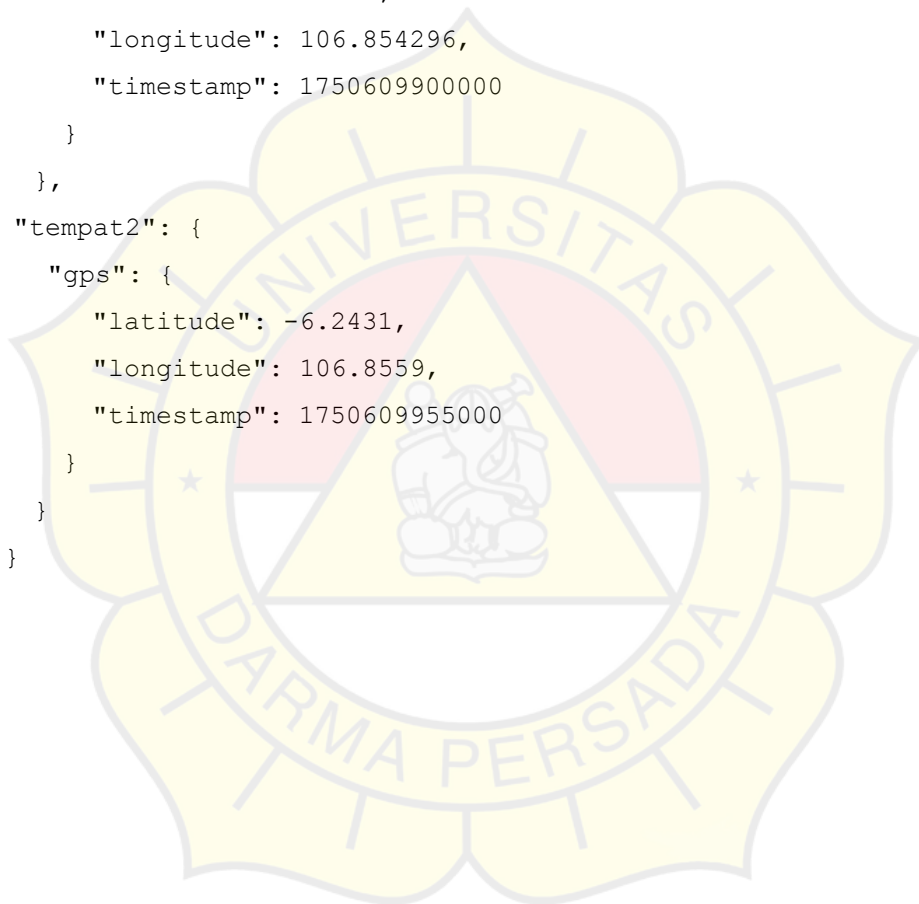
    implementation("com.google.android.gms:play-services-
        maps:18.2.0")
    implementation("com.google.android.gms:play-services-
        location:21.0.1")
    implementation(libs.firebase.database)

    testImplementation("junit:junit:4.13.2")
    androidTestImplementation("androidx.test.ext:junit:1.1
        .5")
    androidTestImplementation("androidx.test.espresso:espr
        esso-core:3.5.1")
}
```

Source Code Database pada Firebase

1. db.json

```
{
  "sampah": {
    "tempat1": {
      "gps": {
        "latitude": -6.242395,
        "longitude": 106.854296,
        "timestamp": 1750609900000
      }
    },
    "tempat2": {
      "gps": {
        "latitude": -6.2431,
        "longitude": 106.8559,
        "timestamp": 1750609955000
      }
    }
  }
}
```



Surat Keterangan Hasil Pengecekan Turnitin



**UNIVERSITAS DARMA PERSADA
UPT PERPUSTAKAAN**

Gedung Rektorat Lantai 3,
Jl.Taman Malaka Selatan, Pondok Kelapa – Jakarta Timur 13450

**SURAT KETERANGAN
HASIL PENGECEKAN TURNITIN**

UPT Perpustakaan Universitas Darma Persada menerangkan telah selesai melakukan pemeriksaan duplikasi/*similarity* menggunakan perangkat lunak Turnitin terhadap hasil karya sebagai berikut:

Judul : Smartwaste Management System berbasis IoT terintegrasi dengan google maps

Penulis : Fiqri Setiyana Agung

NIM : 2018230211

Tgl pemeriksaan : 28 Juli 2025

Dengan hasil Tingkat Kesamaan (*similarity index*) 14%

Demikian Surat Keterangan kami buat, untuk dipergunakan sebagaimana mestinya.

Jakarta, 28 Juli 2025

Ka.UPT Perpustakaan Unsada

Yus Rusmiyati, SS., MM

Batas maksimal similarity 30% untuk Fakultas Sastra dan Ekonomi
Batas maksimal similarity 25% untuk Fakultas Teknik, Kelautan dan Pasca Sarjana

2018230211_Fiqri Setiyana Agung_Smartwaste Management System berbasis IoT terintegrasi dengan google maps

ORIGINALITY REPORT

14%	13%	8%	%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	journal.isas.or.id Internet Source	1%
2	docplayer.info Internet Source	1%
3	ejournal.upm.ac.id Internet Source	1%
4	repository.ulb.ac.id Internet Source	1%
5	dinarek.unsoed.ac.id Internet Source	1%
6	nusabot.id Internet Source	<1%
7	Riska Ayu Damayanti, Lori Trimurti Negari, Samsul Jihad Purnama, Shofwatunnida' Septarini, Muhammad Saiful. "Rancang Bangun Tempat Sampah Otomatis Berbasis Arduino Uno dengan Sensor Ultrasonik", Jurnal PRINTER: Jurnal Pengembangan Rekayasa Informatika dan Komputer, 2024 Publication	<1%
8	www.moreware.org Internet Source	<1%