

LAMPIRAN

Listing Code

```
#include <SoftwareSerial.h>
#include <TinyGPSPlus.h>

TinyGPSPlus gps;
SoftwareSerial gpsSerial(5, 4); // GPS: TX ke 5, RX ke 4
SoftwareSerial sim800l(7, 8); // GSM: TX ke 7, RX ke 8

// === Variabel global ===
String currentLine = "";
String senderNumber = "";
bool waitingForMessage = false;

// === Fitur lokasi otomatis ===
unsigned long previousMillisNormal = 0;
const long intervalNormal = 3600000; // 1 jam (3600.000 ms)
String nomorOtomatis = "+6289676506990";

// === Fitur darurat ===
bool daruratAktif = false;
unsigned long previousMillisDarurat = 0;
const long intervalDarurat = 30000; // 30 detik

// === Buzzer (alarm) ===
#define BUZZER_PIN 3
bool buzzerActive = false;
unsigned long buzzerStartTime = 0;
unsigned long buzzerPreviousMillis = 0;
const long buzzerInterval = 500; // 0,5 detik hidup-mati
bool buzzerState = LOW;
const long buzzerDuration = 20000; // 20 detik

// === Setup ===
void setup() {
  Serial.begin(9600);
  gpsSerial.begin(9600);
  sim800l.begin(9600);

  pinMode(BUZZER_PIN, OUTPUT);
  digitalWrite(BUZZER_PIN, LOW);

  Serial.println("Memulai GPS Tracker...");

  delay(1000);
  sim800l.listen();
```

```

sim800l.println("AT");
delay(500);
sim800l.println("AT+CMGF=1"); // Mode teks
delay(500);
sim800l.println("AT+CNMI=1,2,0,0,0"); // SMS langsung tampil
delay(500);

Serial.println("GSM dan GPS siap.");
}

// === Loop utama ===
void loop() {
    sim800l.listen();

    // === Cek SMS masuk ===
    while (sim800l.available()) {
        char c = sim800l.read();
        Serial.print(c);
        currentLine += c;

        if (c == '\n') {
            if (currentLine.indexOf("+CMT:") >= 0) {
                int startPos = currentLine.indexOf("\"") + 1;
                int endPos = currentLine.indexOf("\"", startPos);
                senderNumber = currentLine.substring(startPos, endPos);
                waitingForMessage = true;
            }
            else if (waitingForMessage && currentLine.length() > 1) {
                String messageContent = currentLine;
                messageContent.trim();

                Serial.println("Isi Pesan: " + messageContent + "");
                prosesPerintah(messageContent, senderNumber);

                waitingForMessage = false;
            }
            currentLine = "";
        }
    }

    // === Cek input dari Serial Monitor ===
    if (Serial.available()) {
        String serialCommand = Serial.readStringUntil('\n');
        serialCommand.trim();
        if (serialCommand.length() > 0) {
            prosesPerintah(serialCommand, nomorOtomatis);
        }
    }
}

```

```

unsigned long currentMillis = millis();

// === Kirim lokasi otomatis tiap 1 jam ===
if (currentMillis - previousMillisNormal >= intervalNormal) {
    previousMillisNormal = currentMillis;
    kirimLokasiKe(nomorOtomatis);
}

// === Kirim lokasi darurat tiap 30 detik ===
if (daruratAktif && (currentMillis - previousMillisDarurat >= intervalDarurat))
{
    previousMillisDarurat = currentMillis;
    kirimLokasiKe(senderNumber);
}

// === Alarm buzzer bip-bip 20 detik ===
if (buzzerActive) {
    if (currentMillis - buzzerPreviousMillis >= buzzerInterval) {
        buzzerPreviousMillis = currentMillis;
        buzzerState = !buzzerState;
        digitalWrite(BUZZER_PIN, buzzerState);
    }
    if (currentMillis - buzzerStartTime >= buzzerDuration) {
        digitalWrite(BUZZER_PIN, LOW);
        buzzerActive = false;
        Serial.println("Alarm buzzer MATI otomatis setelah 20 detik.");
    }
}

delay(100);
}

// === Fungsi untuk memproses perintah ===
void prosesPerintah(String messageContent, String nomorPengirim) {
    if (messageContent.equalsIgnoreCase("lokasi")) {
        kirimLokasiKe(nomorPengirim);
    }
    else if (messageContent.equalsIgnoreCase("darurat")) {
        daruratAktif = true;
        previousMillisDarurat = millis();
        senderNumber = nomorPengirim;
        Serial.println("Mode DARURAT aktif.");
    }
    else if (messageContent.equalsIgnoreCase("off darurat")) {
        daruratAktif = false;
        Serial.println("Mode DARURAT nonaktif.");
    }
}

```

```

else if (messageContent.equalsIgnoreCase("alarm")) {
    buzzerActive = true;
    buzzerStartTime = millis();
    buzzerPreviousMillis = millis();
    buzzerState = HIGH;
    digitalWrite(BUZZER_PIN, buzzerState);
    Serial.println("Alarm buzzer AKTIF (mode bip-bip 20 detik).");
}
else if (messageContent.equalsIgnoreCase("alarm off")) {
    digitalWrite(BUZZER_PIN, LOW);
    buzzerActive = false;
    Serial.println("Alarm buzzer dimatikan manual.");
}
}

// === Fungsi kirim lokasi ===
void kirimLokasiKe(String nomorTujuan) {
    Serial.println("Menyiapkan lokasi GPS...");

    gpsSerial.listen();
    unsigned long timeout = millis() + 5000;
    while (millis() < timeout) {
        while (gpsSerial.available()) {
            gps.encode(gpsSerial.read());
        }
        if (gps.location.isUpdated()) break;
    }

    if (gps.location.isValid()) {
        double lat = gps.location.lat();
        double lng = gps.location.lng();
        String mapsLink = "https://www.google.com/maps/search/?api=1&query=" +
String(lat, 6) + "," + String(lng, 6);
        kirimSMSKe(nomorTujuan, mapsLink);
    } else {
        kirimSMSKe(nomorTujuan, "Lokasi belum tersedia. Coba lagi sebentar lagi.");
    }
}

// === Fungsi kirim SMS ===
void kirimSMSKe(String nomor, String text) {
    sim800l.listen();
    delay(500);
    sim800l.println("AT+CMGF=1");
    delay(500);
    sim800l.print("AT+CMGS=\""");
    sim800l.print(nomor);
    sim800l.println("\n");
}

```

```
delay(500);  
sim800l.print(text);  
delay(500);  
sim800l.write(26); // CTRL+Z  
Serial.println("SMS terkirim ke " + nomor + ": " + text);  
}
```