

SAMPUL



**Sistem Monitoring
Kesehatan Struktur Flyover
Menggunakan MPU6050 Via
Googlesheets.**

**Flyover Structure Health
Monitoring system using
MPU6050 Via GoogleSheets.**

Aflaq Puji Asmoro, ST
NIM 211020100023

Dosen Pembimbing
Shazana Dhiya Ayuni, S.ST. MT

Dosen Penguji
Agus Hayatal Falah, ST., MT.
Dr. Ir. Jamaaluddin, MM.

**Program Studi
Fakultas
Universitas Muhammadiyah Sidoarjo
Maret, 2025**

LEMBAR PENGESAHAN

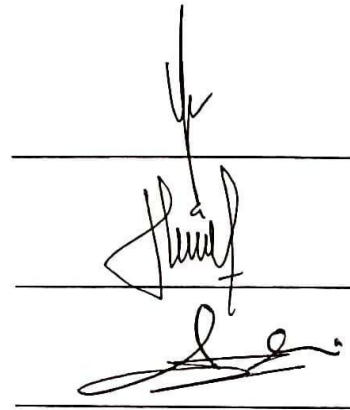
Judul : Sistem Monitoring Kesehatan Struktur Flyover Sumokali Menggunakan
MPU6050 Via GoogleSheet
Nama Mahasiswa : Aflaq Puji Asmoro
NIM : 211020100023

Disetujui oleh

Dosen Pembimbing
(Shazana Dhiya Ayuni, S.ST. MT)

Dosen Penguji 1
(Agus Hayatal Falah, ST., MT.)

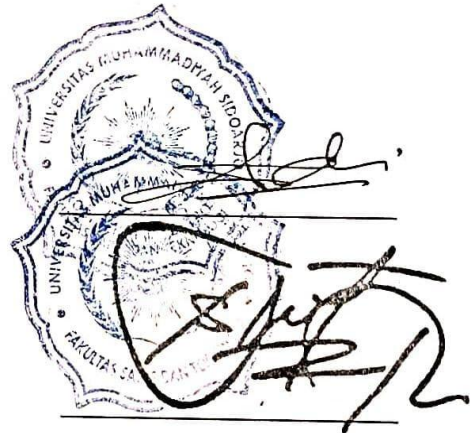
Dosen Penguji 2
(Dr. Ir Jamaaluddin, MM.)



Diketahui oleh

Plt Ketua Program Studi Teknik
Elektro
(Dr. Ir Jamaaluddin, MM.)
NIP/NIK. 214327

Dekan
(Iswanto, St., M.Mt.)
NIP/NIK. 207319



Tanggal Ujian
(21/05/2025)

Tanggal Lulus
(/ /)

DAFTAR ISI

LEMBAR PENGESAHAN.....	2
DAFTAR ISI.....	3
SURAT PERNYATAAN PUBLIKASI ILMIAH.....	5
PERNYATAAN MENGENAI KARYA TULIS ILMIAH DAN SUMBER INFORMASI SERTA PELIMPAHAN HAK CIPTA.....	6
Definisi Singkat.....	7
Node transmitter.....	7
Kegunaan node transmitter.....	7
Cara penggunaan node transmitter.....	7
Program komputer node transmitter.....	8
Node Receiver.....	9
Kegunaan node receiver.....	9
Cara penggunaan node receiver.....	9
Program komputer node receiver.....	10

SURAT PERNYATAAN PUBLIKASI ILMIAH

Yang bertanda tangan dibawah ini, saya:

Nama Mahasiswa : Aflaq Puji Asmoro
NIM : 211020100023
Program Studi : Teknik Elektro
Fakultas : Sains dan Teknologi

DAN

Dosen Pembimbing : Shazana Dhiya Ayuni, S.ST. MT
NIDN : 071209103
Program Studi : Teknik Elektro
Fakultas : Sains dan Teknologi

MENYATAKAN bahwa, karya tulis ilmiah dengan rincian:

Judul : Sistem Monitoring Kesehatan Struktur Flyover Sumokali Menggunakan MPU Via Google Sheets.
Kata Kunci : Program Komputer

TELAH:

1. Disesuaikan dengan petunjuk penulisan di Universitas Muhammadiyah Sidoarjo. Berdasarkan Surat Keputusan Rektor UMSIDA tentang Pedoman Karya Tulis Ilmiah Mahasiswa.
2. Lolos uji cek kesamaan sesuai ketentuan yang berlaku di Universitas Muhammadiyah Sidoarjo.

SERTA*:

- ☐ **Bertanggung jawab untuk** melakukan publikasi karya tulis ilmiah tersebut ke jurnal ilmiah/prosiding sesuai ketentuan Surat Keputusan Rektor UMSIDA tentang Pedoman Karya Tulis Ilmiah. Khususnya Lampiran Huruf B.
- ☒ **Menyerahkan tanggung jawab untuk** melakukan publikasi karya tulis ilmiah tersebut ke jurnal ilmiah/prosiding sesuai ketentuan Surat Keputusan Rektor UMSIDA tentang Pedoman Karya Tulis Ilmiah. Khususnya Lampiran Huruf B kepada Bidang Pengembangan Publikasi Ilmiah DRPM UMSIDA.

Demikian pernyataan dari saya, untuk dipergunakan sebagaimana mestinya. Terima Kasih

Menyetujui,
Dosen Pembimbing

(Shazana Dhiya Ayuni, S.ST. MT)
NIDN.071209103

Sidoarjo, (21/05/2025)
Mahasiswa


(Aflaq Puji Asmoro)
NIM. 211020100023

*Centang salah satu.

PERNYATAAN MENGENAI KARYA TULIS ILMIAH DAN SUMBER INFORMASI SERTA PELIMPAHAN HAK CIPTA

Dengan ini saya menyatakan bahwa karya tulis ilmiah tugas akhir saya dengan judul **“Monitoring Sistem kesehatan struktur flyover sumokali menggunakan mpu 6050 via google sheets.”** adalah karya saya dengan arahan dari dosen pembimbing dan belum diajukan dalam bentuk apapun kepada perguruan tinggi mana pun. Sumber informasi yang berasal atau dikutip dari karya yang diterbitkan maupun tidak diterbitkan dari penulis lain telah disebutkan dalam teks dan dicantumkan dalam Daftar Pustaka di bagian akhir karya tulis ilmiah tugas akhir saya ini.

Dengan ini saya melimpahkan hak cipta dari karya tulis saya kepada Universitas Muhammadiyah Sidoarjo

Sidoarjo, 21 Mei 2025



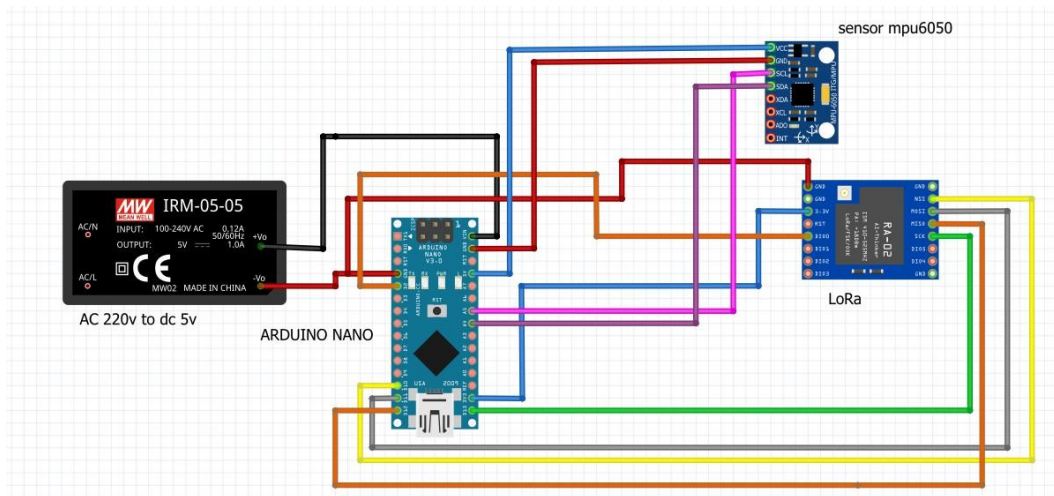
Aflaq Puji Asmoro

NIM.211020100023

Definisi Singkat

Program ini digunakan untuk memonitoring kesehatan struktur flyover sumokali menggunakan sensor mpu 6050 dan akan ditransmisikan oleh lora sx1278 sebagai node receiver dan diolah menggunakan esp32 sebagai mikrokontroler pada receiver dan data akan dibaca secara realtime pada googlesheets dan apabila terjadi nilai melebihi batas atas yang telah ditentukan maka akan muncul notifikasi melalui telegram mengenai tanda bahaya dari flyover sumokali.

Node transmitter



Kegunaan node transmitter

Node transmitter berfungsi sebagai pembaca data oleh sensor dan akan di transmisikan ke node receiver

Node transmitter disini terdiri dari

- Arduino Nano :
berfungsi sebagai mikrokontroler utama pada node transmitter dimana data yang terbaca oleh sensor mpu 6050 akan diolah dan akan di kirimkan ke lora sx1278 untuk ditransmisikan ke node receiver
- Sensor Mpu 6050
Sensor mpu 6050 berfungsi sebagai sensor pembaca gyroscope dan axelerometer pada objek seperti flyover sumokali
- Lora sx1278
Lora sx1278 berfungsi sebagai transmisikan data yang telah diterima dan diolah oleh arduino ke node receiver
- Coverter ac to dc

Digunakan sebagai sumber tegangan utama pada sistem rangkaian ini

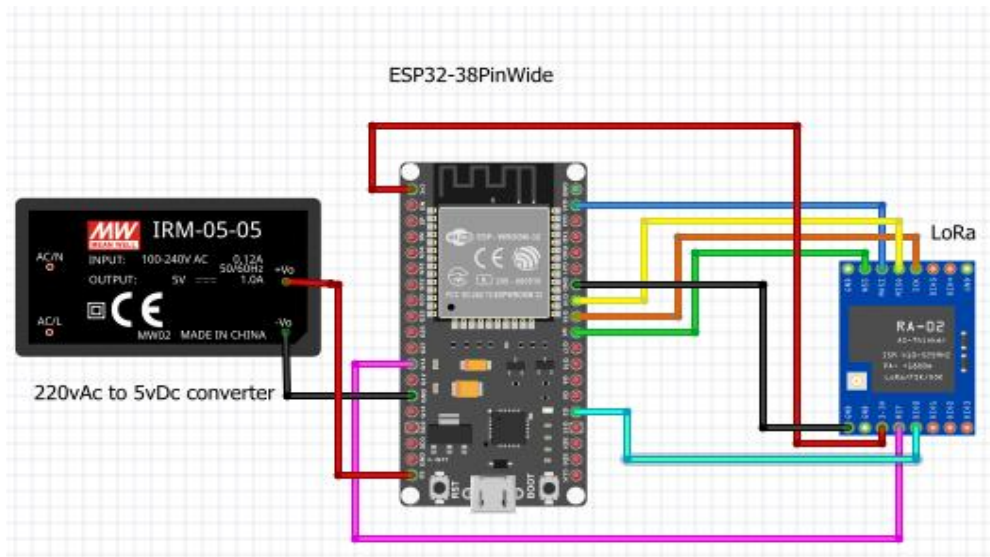
Cara penggunaan node transmitter

- Sambungkan converter ac to dc ke sumber tegangan utama
- Letakan node transmitter kepada objek yang akan diteliti atau dimonitoring, dalam hal ini seperti fly over sumokali
- Sambungkan kabel usb dari arduino ke laptop, kemudian buka software arduino ide dan upload program tersebut
- Setelah upload program tunggu hingga muncul tulisan done uploading, dan cek serial monitor apakah sensor sudah melakukan pembacaan atau belum.

Program komputer node transmitter

```
1  #include <SPI.h>
2  #include <LoRa.h>
3  #include <Adafruit_MPU6050.h>
4  #include <Adafruit_Sensor.h>
5  #include <Wire.h>
6
7  Adafruit_MPU6050 mpu;
8  int counter = 0;
9
10 void setup() {
11   Serial.begin(115200);
12   while (!Serial);
13   Serial.println("LoRa Sender with MPU6050");
14
15   LoRa.setPins(10, 9, 2); //NSS, RST, DIO0
16   if (!LoRa.begin(433E6)) {
17     Serial.println("LoRa init failed!");
18     while (1);
19   }
20   LoRa.setSyncWord(0xF3);
21   Serial.println("LoRa init success!");
22
23   if (!mpu.begin()) {
24     Serial.println("Failed to find MPU6050 chip");
25     while (1) {
26       delay(10);
27     }
28   }
29   Serial.println("MPU6050 Found!");
30
31   mpu.setAccelerometerRange(MPU6050_RANGE_16_G);
32   mpu.setGyroRange(MPU6050_RANGE_2000_DEG);
33   mpu.setGyroRange(MPU6050_RANGE_2000_DEG);
34   mpu.setFilterBandwidth(MPU6050_BAND_5_HZ);
35
36   delay(100);
37 }
38
39 void loop() {
40   sensors_event_t a, g, temp;
41   mpu.getEvent(&a, &g, &temp);
42
43   Serial.print("Sending packet: ");
44   Serial.println(counter);
45
46   LoRa.beginPacket();
47
48   LoRa.print("PKT#");
49   LoRa.print(counter);
50   LoRa.print("|");
51
52   LoRa.print("ACC:");
53   LoRa.print(a.acceleration.x);
54   LoRa.print(",");
55   LoRa.print(a.acceleration.y);
56   LoRa.print(",");
57   LoRa.print(a.acceleration.z);
58   LoRa.print("|");
59
60   LoRa.print("GYR:");
61   LoRa.print(g.gyro.x);
62   LoRa.print(",");
63   LoRa.print(g.gyro.y);
64   LoRa.print(",");
65   LoRa.print(g.gyro.z);
66   LoRa.print("|");
67
68   LoRa.endPacket();
69
70   Serial.print("ACC- X: ");
71   Serial.print(a.acceleration.x);
72   Serial.print(", Y: ");
73   Serial.print(a.acceleration.y);
74   Serial.print(", Z: ");
75   Serial.print(a.acceleration.z);
76   Serial.println(" m/s^2");
77
78   Serial.print("GYR- X: ");
79   Serial.print(g.gyro.x);
80   Serial.print(", Y: ");
81   Serial.print(g.gyro.y);
82   Serial.print(", Z: ");
83   Serial.print(g.gyro.z);
84   Serial.println(" rad/s");
85
86   counter++;
87   delay(5000); //Delay mengirim ke Lora Reciever (ESP32)
```

Node Receiver



Kegunaan node receiver

Node receiver berfungsi sebagai penerima data dari node transmitter dan data akan diterima oleh node receiver dan akan di teruskan ke googlesheets sebagai output dan ke telegram sebagai notifikasi.

Node receiver terdiri dari :

- Esp 32 :
berfungsi sebagai mikrokontroller utama pada node receiver dimana data yang diterima oleh lora sx1278 akan ditampilkan ke googlesheets dan notifikasi bahaya akan ditampilkan dengan telegram bot.
- Lora sx1278
Lora sx1278 berfungsi sebagai penerima data dari node transmitter
- Coverter ac to dc
Digunakan sebagai sumber tegangan utama pada sistem rangkaian ini.

Cara penggunaan node receiver

- Sambungkan kabel usb dari esp32 ke laptop, kemudian buka software arduino ide dan upload program tersebut
- Setelah upload program tunggu hingga muncul tulisan done uploading, dan pastikan apakah lora sx1278 sudah terkoneksi dengan baik dengan munculnya keterangan lora init suksesfull.
- Setelah itu buka googlesheets dan monitoring nilai yang telah dibaca oleh sensor mpu 6050, jika ada nilai yang melebihi dari batas atas yang telah ditentukan, maka secara otomatis akan muncul notifikasi ke telegram bot.
- Apabila itu terjadi bisa menghubungi pihak terkait untuk dilakukan pengecekan.

Program komputer node receiver

```
1  #include <SPI.h>
2  #include <LoRa.h>
3  #include <WiFi.h>
4  #include <ESP_Google_Sheet_Client.h>
5  #include <UniversalTelegramBot.h>
6  #include <WiFiClientSecure.h>
7  #include "time.h"
8
9  // Definisi variabel global Wifi, Google Spreadsheet
10 #define WIFI_SSID "Aku"
11 #define WIFI_PASSWORD "12345678"
12 #define PROJECT_ID "iot-aflaq"
13 #define CLIENT_EMAIL "iot-aflaq@iot-aflaq.iam.gserviceaccount.com"
14
15 const char PRIVATE_KEY[] PROGMEM = "-----BEGIN PRIVATE KEY-----\nMIIEVQIBADANBgkqhkiG9w0BAQEFAASCBCwggSjAgEAAoIBAQCnrDgyt7YFw8uv\nnmLTIwDfy3/4Mu1Q8zXj
16 const char spreadsheetId[] = "1qx9avxjhjAG0gftnMsLzebGU_oc2UBIAfumeoFhTauU";
17
18 //Definisi Variabel Bot Telegram
19 #define BOT_TOKEN "7069587056:AAEzyxkMpn1alQmsrcKr6CEmdbrvX0EACfI"
20 #define CHAT_ID "6650656156"
21
22 WiFiClientSecure secured_client;
23 UniversalTelegramBot bot(BOT_TOKEN, secured_client);
24
25 //Variabel RTC
26 const char ntpServer = "pool.ntp.org";
27 const char backup_ntpServer = "time.google.com";
28 const long gmtoffset_sec = 7 * 3600;
29 const int daylightoffset_sec = 0;
30 struct tm timeinfo;
31
32 // Definisi struct SensorStatus dengan penambahan debug info
33 struct SensorStatus {
34     String accX;
35     String accY;
36     String accZ;
37     String gyrX;
38     String gyrY;
39     String gyrZ;
40     String ket; // Added field for percentage warning
41     float warningPercentage; // Added to store the actual percentage
42
43     // Debug info fields remain the same
44     float rawAccX;
45     float rawAccY;
46     float rawAccZ;
47     float avgAccX;
48     float avgAccY;
49     float avgAccZ;
50
51     SensorStatus() {
52         accX = "NORMAL";
53         accY = "NORMAL";
54         accZ = "NORMAL";
55         gyrX = "NORMAL";
56         gyrY = "NORMAL";
57         gyrZ = "NORMAL";
58         ket = "Normal: 0%";
59         warningPercentage = 0.0;
60         rawAccX = 0.0;
61         rawAccY = 0.0;
62         rawAccZ = 0.0;
63         avgAccX = 0.0;
64         avgAccY = 0.0;
65         avgAccZ = 0.0;
66     }
67
68     void calculateWarningPercentage() {
69         int warningCount = 0;
70         if (accX == "WASPADA") warningCount++;
71         if (accY == "WASPADA") warningCount++;
72         if (accZ == "WASPADA") warningCount++;
73         if (gyrX == "WASPADA") warningCount++;
74         if (gyrY == "WASPADA") warningCount++;
75         if (gyrZ == "WASPADA") warningCount++;
76
77         warningPercentage = (warningCount * 100.0) / 6.0;
78         ket = "Waspada Retak: " + String(warningPercentage, 1) + "%";
79     }
80
81     String getCombinedStatus() {
82         String result = "";
83         result += "ACC:\n";
84         result += "1. X: " + accX + "\n";
85         result += "2. Y: " + accY + "\n";
86         result += "3. Z: " + accZ + "\n";
87         result += "GYR:\n";
88         result += "1. X: " + gyrX + "\n";
89         result += "2. Y: " + gyrY + "\n";
90         result += "3. Z: " + gyrZ + "\n";
91         return result;
92     }
93
94     // Tambahan method untuk debug info
95     String getDebugInfo() {
96         String debug = "";
```

```

96     String debug = "";
97     debug += "Raw ACC - X: " + String(rawAccX) + ", Y: " + String(rawAccY) + ", Z: " + String(rawAccZ) + "\n";
98     debug += "Avg ACC - X: " + String(avgAccX) + ", Y: " + String(avgAccY) + ", Z: " + String(avgAccZ) + "\n";
99     return debug;
100 }
101 };
102
103 // Konstanta threshold yang direvisi
104 const struct {
105     struct {
106         float xMin = -1.5;
107         float xMax = 1.5;
108         float yMin = -2.0;
109         float yMax = 1.0;
110         float zMin = -1.0;
111         float zMax = 8.0;
112     } acc;
113
114     struct {
115         float min = -10.0;
116         float max = 10.0;
117     } gyr;
118 } THRESHOLDS;
119
120 // Buffer untuk moving average
121 const int WINDOW_SIZE = 10;
122 // Tambahkan variabel global yang hilang
123 unsigned long lastPacketTime = 0;
124 unsigned long packetsReceived = 0;
125 unsigned long packetsLost = 0;
126 int lastPacketNum = -1;
127 struct SensorBuffer {
128     float values[WINDOW_SIZE];
129     float values[WINDOW_SIZE];
130     int index;
131     float sum;
132
133     SensorBuffer()
134     : index(0), sum(0.0) {
135         for (int i = 0; i < WINDOW_SIZE; i++) {
136             values[i] = 0.0;
137         }
138     }
139
140     void add(float value) {
141         sum -= values[index];
142         values[index] = value;
143         sum += value;
144         index = (index + 1) % WINDOW_SIZE;
145     }
146
147     float getAverage() {
148         return sum / WINDOW_SIZE;
149     }
150 } accXBuffer, accYBuffer, accZBuffer;
151
152 void tokenStatusCallback(TokenInfo info) {
153     if (info.status == token_status_error) {
154         GSheet.printf("Token info: type = %s, status = %s\n",
155             GSheet.getTokenType(info).c_str(),
156             GSheet.getTokenStatus(info).c_str());
157         GSheet.printf("Token error: %s\n",
158             GSheet.getTokenError(info).c_str());
159     } else {
160         GSheet.printf("Token info: type = %s, status = %s\n",
161             GSheet.getTokenType(info).c_str(),
162             GSheet.getTokenStatus(info).c_str());
163     }
164 }
165
166 string getFormattedTimestamp() {
167     if (!getLocalTime(&timeinfo)) {
168         Serial.println("Failed to obtain time");
169         return "Error getting time";
170     }
171
172     char timeStringBuff[25];
173     strftime(timeStringBuff, sizeof(timeStringBuff), "%d/%m/%Y %H:%M:%S", &timeinfo);
174
175     Serial.print("Current time: ");
176     Serial.println(timeStringBuff);
177     return String(timeStringBuff);
178 }
179
180 void setupTime() {
181     // Konfigurasi waktu dengan multiple server
182     configTime(gmtOffsetSec, daylightOffsetSec, "id.pool.ntp.org", "pool.ntp.org");
183
184     // Tunggu sampai waktu tersinkronisasi
185     int retry = 0;
186     Serial.println("Waiting for time sync...");
187     while (!getLocalTime(&timeinfo) && retry < 10) {
188         Serial.print(".");
189         delay(1000);
190         retry++;
191     }
192     if (getLocalTime(&timeinfo)) {

```

```

192 it (getLocalTime(&timeinfo)) {
193     Serial.println("\nTime synchronized!");
194     char timeStringBuff[50];
195     strftime(timeStringBuff, sizeof(timeStringBuff), "%d/%m/%Y %H:%M:%S", &timeinfo);
196     Serial.printf("Current time: %s\n", timeStringBuff);
197 } else {
198     Serial.println("\nFailed to sync time after retries!");
199 }
200 }
201
202 void setup() {
203     Serial.begin(115200);
204     while (!Serial)
205         ;
206
207     Serial.println("LoRa Receiver with Google Sheets Integration");
208     configTime(gmtOffset_sec, daylightOffset_sec, ntpServer);
209     Wifi.setAutoReconnect(true);
210     Wifi.begin(WIFI_SSID, WIFI_PASSWORD);
211     Serial.print("Connecting to WiFi");
212     while (Wifi.status() != WL_CONNECTED) {
213         Serial.print(".");
214         delay(500);
215     }
216     Serial.println("\nWiFi connected");
217     setupTime();
218
219     GSHEET.printf("ESP Google Sheet Client v%s\n\n", ESP_GOOGLE_SHEET_CLIENT_VERSION);
220     GSHEET.setTokenCallback(tokenStatusCallback);
221     GSHEET.setPrerefreshSeconds(10 * 60);
222     GSHEET.begin(CLIENT_EMAIL, PROJECT_ID, PRIVATE_KEY);
223
224     LoRa.setPins(5, 14, 2); //NSS, RST, DIO0
225     while (!LoRa.begin(433E6)) {
226         Serial.println("LoRa init failed. Retrying...");
227         delay(100);
228     }
229
230     LoRa.setSyncWord(0xF3);
231     LoRa.enableCrc();
232     LoRa.receive();
233
234     Serial.println("LoRa init success!");
235     accXBuffer = SensorBuffer();
236     accYBuffer = SensorBuffer();
237     accZBuffer = SensorBuffer();
238     secured_client.setCACert(TELEGRAM_CERTIFICATE_ROOT);
239 }
240
241 void processData(float accX, float accY, float accZ, float gyrX, float gyrY, float gyrZ, SensorStatus &status) {
242     // Existing buffer updates and calculations remain the same...
243     accXBuffer.add(accX);
244     accYBuffer.add(accY);
245     accZBuffer.add(accZ);
246
247     float xAvg = accXBuffer.getAverage();
248     float yAvg = accYBuffer.getAverage();
249     float zAvg = accZBuffer.getAverage();
250
251     status.rawAccX = accX;
252     status.rawAccY = accY;
253     status.rawAccZ = accZ;
254     status.avgAccX = xAvg;
255     status.avgAccY = yAvg;
256     status.avgAccZ = zAvg;
257
258     status.avgAccZ = zAvg;
259
260     status.accX = (xAvg > THRESHOLDS.acc.xMax || xAvg < THRESHOLDS.acc.xMin) ? "WASPADA" : "NORMAL";
261     status.accY = (yAvg > THRESHOLDS.acc.yMax || yAvg < THRESHOLDS.acc.yMin) ? "WASPADA" : "NORMAL";
262     status.accZ = (zAvg > THRESHOLDS.acc.zMax || zAvg < THRESHOLDS.acc.zMin) ? "WASPADA" : "NORMAL";
263
264     status.gyrX = (gyrX > THRESHOLDS.gyr.max || gyrX < THRESHOLDS.gyr.min) ? "WASPADA" : "NORMAL";
265     status.gyrY = (gyrY > THRESHOLDS.gyr.max || gyrY < THRESHOLDS.gyr.min) ? "WASPADA" : "NORMAL";
266     status.gyrZ = (gyrZ > THRESHOLDS.gyr.max || gyrZ < THRESHOLDS.gyr.min) ? "WASPADA" : "NORMAL";
267
268     // Calculate warning percentage and update status
269     status.calculateWarningPercentage();
270
271     // Send Telegram alert if warning percentage is 50% or higher
272     if (status.warningPercentage >= 0.0) {
273         String alertMessage = "⚠ PERINGATAN KERETAKAN!\n\n";
274         alertMessage += "Status Sensor:\n" + status.getCombinedStatus() + "\n";
275         alertMessage += "\nPeringatan: " + status.ket;
276         bot.sendMessage(CHAT_ID, alertMessage, "");
277     }
278
279     // Debugging output remains the same...
280     Serial.println(status.getDebugInfo());
281 }
282
283 void sendToGoogleSheets(int packetNum, float accX, float accY, float accZ, float gyrX, float gyrY, float gyrZ, SensorStatus status) {
284     FirebaseJson response;
285     FirebaseJson valueRange;
286     String formattedTime = getFormattedTimestamp();
287
288     valueRange.add("majorDimension", "COLUMNS");

```

```

288 valueRange.add("majorDimension", "COLUMNS");
289 valueRange.set("values/[0]/[0]", formattedTime);
290 valueRange.set("values/[1]/[0]", accX);
291 valueRange.set("values/[2]/[0]", accY);
292 valueRange.set("values/[3]/[0]", accZ);
293 valueRange.set("values/[4]/[0]", gyrX);
294 valueRange.set("values/[5]/[0]", gyrY);
295 valueRange.set("values/[6]/[0]", gyrZ);
296 valueRange.set("values/[7]/[0]", status.getCombinedStatus());
297 valueRange.set("values/[8]/[0]", status.ket);
298
299 Serial.println("\nAppending data to spreadsheet...");
300 Serial.print("timestamp: ");
301 Serial.println(formattedTime);
302
303 bool success = GSheet.values.append(&response, spreadsheetId, "Sheet1!A1", &valueRange);
304 if (success) {
305     Serial.println("Data sent successfully!");
306     response.toString(Serial, true);
307 } else {
308     Serial.println("Failed to send data:");
309     Serial.println(GSheet.errorReason());
310 }
311
312 valueRange.clear();
313 Serial.println();
314 Serial.println(ESP.getFreeHeap());
315 }
316
317 void processPacket(String receivedData) {
318     int packetNum = -1;
319     float accX, accY, accZ;
320     int packetNum = -1;
321     float accX, accY, accZ;
322     float gyrX, gyrY, gyrZ;
323     SensorStatus status; // Changed from String to SensorStatus
324
325     int pktIndex = receivedData.indexOf("PKT#");
326     int accIndex = receivedData.indexOf("ACC:");
327     int gyrIndex = receivedData.indexOf("GYR:");
328
329     if (pktIndex != -1 && accIndex != -1 && gyrIndex != -1) {
330         packetNum = receivedData.substring(pktIndex + 4, accIndex - 1).toInt();
331
332         String accData = receivedData.substring(accIndex + 4, gyrIndex - 1);
333         int comma1 = accData.indexOf(',');
334         int comma2 = accData.lastIndexOf(',');
335         accX = accData.substring(0, comma1).toFloat();
336         accY = accData.substring(comma1 + 1, comma2).toFloat();
337         accZ = accData.substring(comma2 + 1).toFloat();
338
339         String gyrData = receivedData.substring(gyrIndex + 4);
340         comma1 = gyrData.indexOf(',');
341         comma2 = gyrData.lastIndexOf(',');
342         gyrX = gyrData.substring(0, comma1).toFloat();
343         gyrY = gyrData.substring(comma1 + 1, comma2).toFloat();
344         gyrZ = gyrData.substring(comma2 + 1).toFloat();
345
346         processData(accX, accY, accZ, gyrX, gyrY, gyrZ, status);
347
348         if (GSheet.ready()) {
349             sendToGoogleSheets(packetNum, accX, accY, accZ, gyrX, gyrY, gyrZ, status);
350         }
351     }
352
353     // Print debug information
354     Serial.println("\n--- Received Packet ---");
355     Serial.print("Packet #: ");
356     Serial.println(packetNum);
357     Serial.println("Accelerometer data: ");
358     Serial.print("X: ");
359     Serial.print(accX);
360     Serial.print(" | ");
361     Serial.print("Y: ");
362     Serial.print(accY);
363     Serial.print(" | ");
364     Serial.print("Z: ");
365     Serial.print(accZ);
366     Serial.println("Gyroscope data: ");
367     Serial.print("X: ");
368     Serial.print(gyrX);
369     Serial.print(" | ");
370     Serial.print("Y: ");
371     Serial.print(gyrY);
372     Serial.print(" | ");
373     Serial.print("Z: ");
374     Serial.print(gyrZ);
375     Serial.println();
376 }
377
378 void loop() {
379     int packetSize = LoRa.parsePacket();
380     if (packetSize) {
381         String receivedData = "";
382         while (LoRa.available()) {
383             receivedData += (char)LoRa.read();
384         }
385     }
386 }

```

```
354     Serial.println("Accelerometer data: ");
355     Serial.print("X: ");
356     Serial.print(accX);
357     Serial.print(" | ");
358     Serial.print("Y: ");
359     Serial.print(accY);
360     Serial.print(" | ");
361     Serial.print("Z: ");
362     Serial.println(accZ);
363     Serial.println("Gyroscope data: ");
364     Serial.print("X: ");
365     Serial.print(gyrX);
366     Serial.print(" | ");
367     Serial.print("Y: ");
368     Serial.print(gyrY);
369     Serial.print(" | ");
370     Serial.print("Z: ");
371     Serial.println(gyrZ);
372 }
373 }
374
375 void loop() {
376     int packetsize = LoRa.parsePacket();
377     if (packetsize) {
378         String receivedData = "";
379         while (LoRa.available()) {
380             receivedData += (char)LoRa.read();
381         }
382         processPacket(receivedData);
383         LoRa.receive();
384     }
385 }
```

SURAT PENCATATAN CIPTAAN



SURAT PENCATATAN CIPTAAN

Dalam rangka perlindungan ciptaan di bidang ilmu pengetahuan, seni dan sastra berdasarkan Undang-Undang Nomor 28 Tahun 2014 tentang Hak Cipta, dengan ini menerangkan:

Nomor dan tanggal permohonan : EC002025049103, 14 Mei 2025

Pencipta

Nama : Aflaq Puji Asmoro, Shazana Dhiya Ayuni, S.ST., MT. dkk

Alamat : Sumotuwo RT.001 RW.005, Desa Sumorame, Kecamatan Candi, Kabupaten Sidoarjo, Candi, Kab. Sidoarjo, Jawa Timur, 61271

Kewarganegaraan : Indonesia

Pemegang Hak Cipta

Nama : Universitas Muhammadiyah Sidoarjo

Alamat : Jl. Mojopahit 666-B, Sidoarjo, Sidoarjo, Kab. Sidoarjo, Jawa Timur, 61215

Kewarganegaraan : Indonesia

Jenis Ciptaan : Program Komputer

Judul Ciptaan : Sistem Monitoring Kesehatan Struktur Flyover Sumokali Menggunakan MPU 6050 Via Google Sheets

Tanggal dan tempat diumumkan untuk pertama kali di wilayah Indonesia atau di luar wilayah Indonesia : 14 Mei 2025, di Kab. Sidoarjo

Jangka waktu perlindungan : Berlaku selama 50 (lima puluh) tahun sejak Ciptaan tersebut pertama kali dilakukan Pengumuman.

Nomor Pencatatan : 000889364

adalah benar berdasarkan keterangan yang diberikan oleh Pemohon.

Surat Pencatatan Hak Cipta atau produk Hak terkait ini sesuai dengan Pasal 72 Undang-Undang Nomor 28 Tahun 2014 tentang Hak Cipta.



a.n. MENTERI HUKUM
DIREKTUR JENDERAL KEKAYAAN INTELEKTUAL
u.b
Direktur Hak Cipta dan Desain Industri

Agung Damarsasongko, SH., MH.
NIP. 196912261994031001



Disclaimer:

1. Dalam hal pemohon memberikan keterangan tidak sesuai dengan surat pernyataan, Menteri berwenang untuk mencabut surat pencatatan permohonan.
2. Surat Pencatatan ini telah disegel secara elektronik menggunakan segel elektronik yang diterbitkan oleh Balai Besar Sertifikasi Elektronik, Badan Siber dan Sandi Negara.
3. Surat Pencatatan ini dapat dibuktikan keasliannya dengan memindai kode QR pada dokumen ini dan informasi akan ditampilkan dalam browser.