

SISTEM PEMANTAUAN KECEPATAN DAN LOKASI PADA KENDARAAN BUS DENGAN PERINGATAN SUARA DAN NOTIFIKASI TELEGRAM

Fadhil Ramadhan¹, Sanyata Purwidayanta²

Prodi Teknik Informatika, STMIK DCI Tasikmalaya

Email: fadhilramadhan@gmail.com¹, Email: spurwidayan@gmail.com²

ABSTRAK

Penelitian ini merancang sistem pemantauan kecepatan dan lokasi kendaraan bus berbasis IoT menggunakan ESP32 dan sensor GPS Neo-6M untuk mendeteksi posisi serta kecepatan kendaraan secara real-time. Sistem dilengkapi DFPlayer Mini yang memberikan peringatan suara saat kecepatan melebihi batas yang ditentukan, serta mengirimkan notifikasi ke aplikasi Telegram untuk memudahkan pengawasan oleh pengelola. Pengujian yang dilakukan pada kendaraan roda dua dan simulasi pada bus menunjukkan bahwa sistem mampu mendeteksi pelanggaran kecepatan dengan akurat dan mengirimkan notifikasi secara responsif. Dengan demikian, sistem ini dapat menjadi solusi untuk meningkatkan keselamatan, kenyamanan, dan pengawasan operasional kendaraan umum serta mendukung pengembangan konsep smart transportation.

Kata kunci: Internet of Things (IoT), Pemantauan Kecepatan, Lokasi Kendaraan, ESP32, GPS Neo-6M, DFPlayer Mini, Telegram.

I PENDAHULUAN

Transportasi merupakan sarana penting yang digunakan manusia dalam menjalankan aktivitas sehari-hari. Salah satu moda transportasi yang banyak digunakan masyarakat adalah kendaraan umum, khususnya bus. Pada momen tertentu seperti hari raya, peningkatan volume lalu lintas menyebabkan perlunya pengawasan dan pemantauan yang lebih optimal terhadap operasional bus guna meningkatkan keselamatan

dan kenyamanan pengguna transportasi umum.

Salah satu permasalahan pada transportasi umum adalah rendahnya kesadaran pengemudi dalam mematuhi batas kecepatan, yang dapat meningkatkan risiko kecelakaan lalu lintas. Kondisi ini tidak hanya mengurangi kenyamanan dan kepuasan penumpang, tetapi juga membahayakan keselamatan pengguna jalan. Berdasarkan penelitian, pelanggaran batas

kecepatan masih menjadi salah satu penyebab utama terjadinya kecelakaan pada kendaraan umum (Darmawan et al., 2020), sehingga diperlukan solusi yang efektif untuk membantu pengemudi mengendalikan kecepatan kendaraan dan menekan angka kecelakaan.

Untuk mengatasi permasalahan ini, diperlukan sistem pemantauan kecepatan dan lokasi kendaraan yang dapat memberikan notifikasi otomatis apabila sopir melebihi batas kecepatan yang sudah ditentukan. Sistem ini dirancang untuk memberikan peringatan notifikasi melalui suara dan aplikasi dan pelacakan lokasi kendaraan, sehingga kendaraan dapat terpantau dan terkontrol melalui informasi yang terkirim ke bagian kantor atau pool bus.

Berdasarkan permasalahan tersebut penulis tertarik untuk meneliti dan mengembangkan proyek yang berjudul "Sistem Pemantauan Kecepatan Dan Lokasi Kendaraan Berbasis Internet Of Things (IoT)". Dengan adanya sistem ini, diharapkan dapat membantu pengelola armada bus dalam meningkatkan pengawasan operasional, mendorong pengemudi agar lebih disiplin, serta meningkatkan keselamatan dan kenyamanan penumpang secara keseluruhan.

II LANDASAN TEORI

2.1 Pengertian Sistem

Pengertian sistem menurut (Maydianto & Ridho, 2021), sistem adalah jaringan proses kerja yang saling terkait dan berkumpul guna untuk mencapai sebuah tujuan serta melakukan suatu kegiatan.

2.2 Pengertian Pemantauan

Pemantauan adalah tindakan teliti dalam mengamati keadaan atau situasi tertentu dengan tujuan supaya semua fakta atau data yang diperoleh dari observasi tersebut dapat menjadi landasan untuk mengambil keputusan selanjutnya yang dibutuhkan (Ariefni & Legowo, 2018).

2.3 Pengertian kecepatan

Kecepatan kendaraan adalah rata-rata jarak yang dapat ditempuh suatu kendaraan pada suatu ruas jalan dalam satu satuan waktu tertentu. Kecepatan dari suatu kendaraan dipengaruhi oleh faktor-faktor manusia, kendaraan dan prasarana, serta dipengaruhi pula oleh arus lalu lintas, kondisi cuaca dan lingkungan alam sekitarnya (Darmawan et al., 2020).

2.4 Pengertian Lokasi

Lokasi adalah tempat atau letak suatu objek di permukaan bumi. Lokasi kendaraan didefinisikan sebagai posisi geografis suatu kendaraan pada waktu tertentu yang diidentifikasi melalui koordinat spasial (latitude, longitude) dan atribut lingkungan sekitarnya (Bosar Panjaitan & Irfan Aprian Iswandana, 2023).

2.5 Pengertian Internet Of Things (IoT)

Internet of Things (IoT) adalah sebuah konsep/skenario dimana suatu objek yang memiliki kemampuan untuk mentransfer data melalui jaringan tanpa memerlukan interaksi manusia ke manusia atau manusia ke komputer (Darmawan et al., 2020)

2.6 Pengertian Notifikasi

Notifikasi adalah pemberitahuan yang dikirimkan sistem kepada pengguna melalui email, aplikasi, atau website melalui jaringan internet, jaringan seluler, dan alat yang dapat memberikan pemberitahuan kepada pengguna sistem atau aplikasi (Mubyarto & Awaloka, 2022).

2.7 NodeMCU ESP32

NodemCU ESP32 adalah perangkat keras yang dapat menjalankan firmware open source yang berfungsi sebagai mikrokontroler.

2.8 GPS Neo 6m

Modul GPS ublox Neo-6M berfungsi sebagai penerima (receiver) sinyal GPS (Global Positioning System Receiver) dalam menangkap dan memproses sinyal dari satelit, sehingga dapat mendeteksi lokasi suatu objek (Rosyady et al., 2022).

2.9 Df Player Mini MP3

DFPlayer Mini MP3 adalah modul pemutar audio berukuran kecil yang dapat memutar file audio dalam format MP3 langsung dari kartu microSD. Modul ini dapat dioperasikan secara mandiri menggunakan tombol-tombol atau dikendalikan melalui mikrokontroler seperti Arduino atau ESP32 menggunakan komunikasi serial.

2.10 Telegram

Telegram adalah aplikasi chatting yang memiliki keunggulan berupa keringanan, kecepatan, dan bebas dari iklan. Aplikasi ini dilengkapi dengan sistem bot, yang juga dikenal sebagai telegram bot, yang dapat

digunakan untuk berkomunikasi dengan perangkat mikrokontroler. Telegram merupakan layanan pesan instan berbasis cloud yang dapat diakses (Waluyo & Putra, 2024).

III ANALISIS MASALAH

3.1 Analisis Sistem

Analisis sistem adalah proses evaluasi yang dilakukan untuk memahami bagaimana sistem yang ada bekerja, mengidentifikasi permasalahan yang terjadi, serta mencari solusi yang tepat guna meningkatkan kinerja sistem yang akan dikembangkan. Dalam penelitian ini, analisis sistem bertujuan untuk memahami kebutuhan pengguna serta mendesain solusi yang tepat dalam membangun sistem monitoring Kendaraan berbasis sensor gps lokasi dan notifikasi telegram.

3.2 Analisis Masalah

Masyarakat dengan pengguna kendaraan angkutan umum sering mengalami beberapa keluhan seperti sopir yang melebihi batas kecepatan. Beberapa permasalahan yang sering terjadi antara lain:

1. Terjadi kecelakaan lalu lintas yang disebabkan oleh kecepatan kendaraan yang melebihi batas.
2. Belum adanya sistem pemantauan atau monitoring secara otomatis pada kendaraan angkutan umum.
3. Tidak adanya sistem otomatis yang memberikan peringatan ketika kendaraan melebihi batas kecepatan.

3.3 Analisis Kebutuhan

Sistem Pemantau kecepatan dan lokasi ini dirancang untuk memonitoring dan melacak kendaraan secara real time yang melebihi batas kecepatan yang sudah ditetapkan. Sistem memanfaatkan sensor GPS Neo 6m untuk melacak lokasi dan juga menghitung kecepatan pada kendaraan.

Komponen utama yang digunakan dalam sistem ini meliputi:

1. Mikrokontroler ESP32 untuk pengolahan data.
2. Sensor GPS Neo-6m untuk mendeteksi kecepatan dan lokasi pada kendaraan.
3. Tampilan layar Lcd Oled untuk menampilkan informasi kecepatan seperti Speedometer
4. DF mini player untuk peringatan suara melalui speaker.
5. Platform aplikasi telegram untuk kontrol jarak jauh dan monitoring.
6. Data informasi yang diberikan berupa nama sopir, rute bus atau plat nomor kendaraan sebagai kebutuhan notifikasi

Data dari sensor diproses secara real-time oleh ESP32, sistem menentukan lokasi pada kendaraan secara real time yang dikirim melalui telegram. Pengguna bisa melihat data, data yang dikirim nya adalah data lokasi, data kecepatan pada kendaraan yang bisa terkontrol dari jarak jauh.

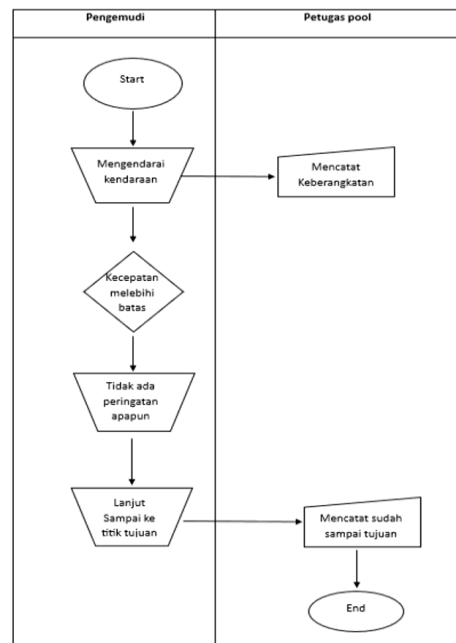
3.4 Analisis Masalah Pengguna

Proses beberapa permasalahan dalam penggunaannya. Berikut adalah penjabaran masalah-masalah tersebut:

1. Tidak ada sistem peringatan otomatis saat melebihi batas kecepatan yang bisa menyebabkan kecelakaan lalu lintas.
2. Petugas pool tidak dapat memantau kecepatan kendaraan yang melanggar secara langsung.
3. Tidak mengetahui lokasi pasti kendaraan jika terjadi insiden.
4. Tidak ada data rekam jejak untuk analisis kecelakaan atau pelanggaran.

3.5 Alur Kerja Sistem

Flowmap Pemantauan Kendaraan Manual



Gambar 3.1 Flowmap pemantauan Kendaraan

IV PERANCANGAN SISTEM

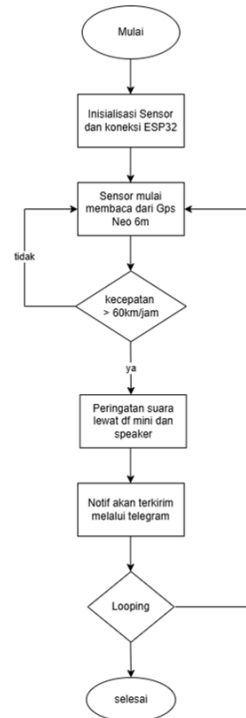
4.1 Alat yang dibutuhkan

Ini adalah daftar peralatan yang diperlukan untuk membangun alat monitoring kendaraan menggunakan sensor kecepatan dan lokasi:

1. NodeMCU ESP32 : Mikrokontroler untuk mengendalikan seluruh sistem.
2. Board NodeMCU ESP32 : papan board untuk beberapa komponen sensor.
3. Modul GPS NEO-6M : digunakan untuk mengetahui lokasi dan data kecepatan kendaraan.
4. Layar Lcd Oled : Untuk menampilkan data kecepatan dan lokasi atau info satelit yang sudah terhubung.
5. DfMini Player : Alat untuk menyimpan data voice suara yang disambungkan melalui speaker mini.
6. Speaker Mini : digunakan untuk mengeluarkan suara atau voice yang sudah tersimpan.
7. Kabel jumper : Untuk koneksi komponen.
8. Kabel USB : Digunakan untuk menyediakan daya dari komputer dan juga untuk mengunggah kode ke NodeMCU ESP32.
9. Arduino IDE : adalah perangkat lunak yang digunakan untuk mengembangkan dan memprogram sketch pada board Arduino yang akan diprogram.
10. Internet Acces : Untuk mengirimkan data.
11. Box Alat : Untuk menyimpan rangkaian yang sudah jadi.
12. Power Bank : Sebagai catu daya untuk menyuplai.
13. Laptop.

4.2 Tahapan Perancangan

4.2.1 Flowchart Sistem Pemantauan Kendaraan Otomatis



Gambar 4.1 Flowchart Sistem Pemantauan Kendaraan Otomatis

Penjelasan proses perancangan alat pemantauan kecepatan dan lokasi kendaraan secara usulan atau sebagai berikut :

1. Proses dimulai ketika ESP32 dinyalakan dan diberi catu daya
2. ESP32 melakukan inisialisai modul GPS Neo-6m, Dfplayer mini dan koneksi wifi untuk notifikasi.
3. ESP32 mencoba menyambung ke jaringan WIFI, jika gagal akan terus mencoba hingga terkoneksi.
4. Sistem akan membaca dan menerima data gps dan kecepatan kendaraan berupa latitude, longitude dan kecepatan km/h.

5. Apabila kecepatan melebihi 60 km/h maka peringatan suara akan berbunyi melalui speaker dan notifikasi akan langsung dikirim melalui aplikasi.

V IMPLEMENTASI

5.1 Implementasi Sistem

Tahapan selanjutnya dari kegiatan analisis dan perancangan sistem yaitu tahap implementasi sistem. Tahapan ini juga merupakan tahap meletakkan system supaya siap untuk di operasikan. Pada bab ini akan diuraikan mengenai implementasi sistem pemantauan kendaraan .

5.2 Perangkat yang Dibutuhkan

5.2.1 perangkat Keras

Perangkat keras yang digunakan untuk menunjang pembuatan perangkat ini adalah sebagai berikut :

1. Nodemcu ESP32
2. Modul Gps Neo 6m
3. Kabel Jumper
4. DF Player Mini
5. Layar LCD Oled
6. Speaker 3W
7. Kabel USB Type C
8. Box Alat
9. Micro SD Card
10. Power Bank
11. Solder

5.2.2 Perangkat Lunak

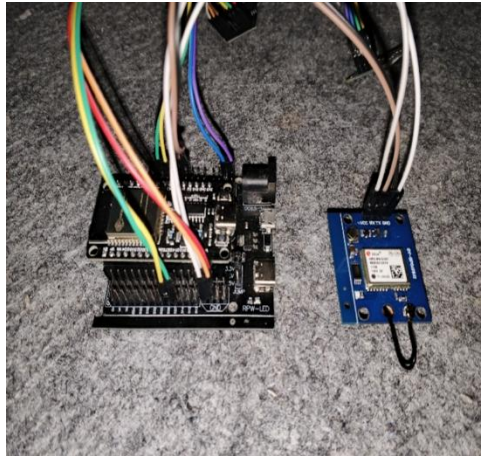
Perangkat keras yang disebutkan sebelumnya, diperlukan juga perangkat lunak untuk pendukung perangkat keras agar bekerja sesuai dengan yang di harapkan.

Perangkat lunak yang dibutuhkan untuk membuat sistem perangkat ini adalah sebagai berikut :

1. Arduino IDE
2. Fritzing
3. MS Word

5.3 Instalasi Perangkat

A. Rangkaian Modul Gps Neo 6m dan ESP32



B. Rangkaian DF player mini dan Speaker Ke ESP32



C. Rangkaian Layar Lcd Oled dan ESP32



D. Rangkaian Keseluruhan sistem dalam box



E. Rangkaian Sistem Keseluruhan Tampak Depan



F. Rangkaian Sistem Keseluruhan Tampak Belakang



5.4 Implementasi Program

```
#include <TinyGPSPlus.h>
#include <HardwareSerial.h>
#include <DFRobotDFPlayerMini.h>
#include <WiFi.h>
#include <HTTPClient.h>
#include <WiFiManager.h>
#include <WiFiClientSecure.h>
#include <Usg2lib.h>
#include <Wire.h>
#include <math.h>

// ===== Konfigurasi OLED =====
U8G2_SSD1306_128X64_NONAME_F_HW_I2C u8g2(U8G2_R0, /* reset=*/
U8Xc_PIN_NONE, 19, I2);

// ===== Konfigurasi Telegram =====
String botToken = "7459226595:AArBsoS_Tkch5wAxqcCfXjiiXxe1SLT4Mc";
String chatID = "-4834906909";

// ===== Hardware Setup =====
TinyGPSPlus gps;
HardwareSerial GPS(1); // RX = 16, TX = 17
HardwareSerial mp3Serial(2); // RX = 26, TX = 27
DFRobotDFPlayerMini mp3;

// ===== Parameter Speedometer =====
const int SCREEN_WIDTH = 128;
const int SCREEN_HEIGHT = 64;
const int CENTER_X = SCREEN_WIDTH / 2;
const int CENTER_Y = 35;
const int RADIUS_Y = 30;
const int NEEDLE_LENGTH = 25;

// ===== Variabel Sistem =====
float speedKmph = 0;
float filteredSpeed = 0;
bool alreadyAlerted = false;
unsigned long lastAlertTime = 0;
const unsigned long alertInterval = 60000; // 60 detik

unsigned long lastDrawTime = 0;
const unsigned long drawInterval = 100; // 100ms

// Variabel Audio Alert
unsigned long lastSoundTime = 0;
const unsigned long soundRepeatInterval = 10000; // 10 detik
bool isPlayingSound = false;
bool firstBoot = true; // Flag boot pertama

// Variabel GPS
int num_sat;
double lat, lon;
String hour, minute;
int second;
String heading;

// ===== Setup Awal =====
void setup() {
  Serial.begin(115200);
  GPS.begin(9600, SERIAL_8N1, 16, 17);

  // Inisialisasi DFPlayer dengan proteksi
  mp3Serial.begin(9600, SERIAL_8N1, 26, 27);
  delay(1000); // Tunggu stabilisasi daya

  if (!mp3.begin(mp3Serial, true, false)) {
    Serial.println("✗ DFPlayer tidak terdeteksi.");
  } else {
    Serial.println("☑ DFPlayer siap.");
    delay(300);
    mp3.volume(30);
    mp3.stop(); // Pastikan tidak ada suara saat startup
  }

  // Koneksi WiFi
  WiFiManager wm;
  if (!wm.autoConnect("ESP32-GPS")) {
    Serial.println("✗ Gagal konek WiFi.");
    ESP.restart();
  }
  Serial.println("☑ WiFi terhubung: " + WiFi.localIP().toString());

  // Inisialisasi OLED
  u8g2.begin();
  u8g2.setFont(u8g2_font_5x7_tf);
  u8g2.setFontRefHeightExtendedText();
  u8g2.setDrawColor(1);
  u8g2.setFontPosTop();
}
```

```

// ===== Loop Utama =====
void loop() {
  // 1. Baca data GPS
  while (GPS.available()) {
    gps.encode(GPS.read());
  }

  // 2. Proses data GPS valid
  if (gps.location.isValid()) {
    lat = gps.location.lat();
    lon = gps.location.lng();
    speedKmph = gps.speed.kmph();
    num_sat = gps.satellites.value();

    if (gps.time.isValid()) {
      hour = pad2(gps.time.hour());
      minute = pad2(gps.time.minute());
      second = gps.time.second();
    }

    heading = gps.course.isValid() ? gps.cardinal(gps.course.value())
    : "---";
  }

  // 3. Filter kecepatan
  filteredSpeed = 0.95 * filteredSpeed + 0.05 * speedKmph; // Filter
  lebih ketat

  // 4. Handle boot pertama
  if (firstBoot) {
    firstBoot = false;
    mp3.stop(); // Pastikan stop saat boot
    delay(100);
  }

  // 5. Update status suara
  updateSoundStatus();

  // 6. Handle alert system dengan histeresis
  handleAlertSystem();

  // 7. Update tampilan
  if (millis() - lastDrawTime > drawInterval) {
    drawSpeedometer();
    lastDrawTime = millis();
  }
}

// ===== Fungsi Sistem =====

void updateSoundStatus() {
  if (isPlayingSound && millis() - lastSoundTime > 2000) {
    isPlayingSound = false;
  }
}

```

```

void handleAlertSystem() {
  // Dengan histeresis 2 km/jam
  if (filteredSpeed > 32.0) { // Threshold + buffer
    if (!alreadyAlerted) {
      mp3.play(1);
      isPlayingSound = true;
      kirimLiveLocation(lat, lon);
      kirimTeksLokasi(lat, lon, filteredSpeed);
      alreadyAlerted = true;
      lastAlertTime = millis();
      lastSoundTime = millis();
    }
    else if (millis() - lastSoundTime > soundRepeatInterval &&
    !isPlayingSound) {
      mp3.play(1);
      isPlayingSound = true;
      lastSoundTime = millis();
    }
  }
  else if (filteredSpeed < 28.0) { // Threshold - buffer
    alreadyAlerted = false;
    isPlayingSound = false;
  }
}

void drawSpeedometer() {
  int needle_angle = map(constrain(filteredSpeed, 0, 200), 0, 200,
  135, 405);

  u8g2.firstPage();
  do {
    // Draw gauge
    u8g2.drawCircle(CENTER_X, CENTER_Y, RADIUS+6,
    U8G2_DRAW_UPPER_RIGHT);
    u8g2.drawCircle(CENTER_X, CENTER_Y, RADIUS+4,
    U8G2_DRAW_UPPER_RIGHT);
    u8g2.drawCircle(CENTER_X, CENTER_Y, RADIUS+6,
    U8G2_DRAW_UPPER_LEFT);
    u8g2.drawCircle(CENTER_X, CENTER_Y, RADIUS+4,
    U8G2_DRAW_UPPER_LEFT);

    // Labels
    u8g2.setFont(u8g2_font_5x7_tf);
    u8g2.drawStr(CENTER_X-10, CENTER_Y-25, "km/h");
    u8g2.drawStr(CENTER_X-15, CENTER_Y-15, "SPEED");
    u8g2.drawStr(15, CENTER_Y+5, "0");
    u8g2.drawStr(100, CENTER_Y+5, "200");

    // Needle
    float x = CENTER_X + NEEDLE_LENGTH * cos(needle_angle * PI / 180);
    float y = CENTER_Y - NEEDLE_LENGTH * sin(needle_angle * PI / 180);
    u8g2.drawLine(CENTER_X, CENTER_Y, x, y);
    u8g2.drawDisc(CENTER_X, CENTER_Y, 3, U8G2_DRAW_ALL);

    // Digital speed
    u8g2.setFont(u8g2_font_profont22_tf);
    u8g2.setCursor(CENTER_X-12, CENTER_Y+10);

```

```

if (filteredSpeed < 10) u8g2.print("0");
u8g2.print((int)filteredSpeed);

// Satellite info
u8g2.setFont(u8g2_font_5x7_tf);
u8g2.setCursor(2, 0);
u8g2.print("SAT:");
u8g2.print(num_sat);

// Time
u8g2.setCursor(90, 0);
u8g2.print(hour);
u8g2.print(":");
u8g2.print(minute);

// Location
u8g2.setCursor(2, 55);
u8g2.print(lat, 4);

// Heading
u8g2.setCursor(100, 55);
u8g2.print(heading);

} while (u8g2.nextPage());
}

// ===== Fungsi Telegram =====

void kirimLiveLocation(float lat, float lon) {
  if (WiFi.status() == WL_CONNECTED) {
    WiFiClientSecure client;
    client.setInsecure();
    HTTPClient http;

    String url = "https://api.telegram.org/bot" + botToken +
      "/sendLocation?chat_id=" + chatID +
      "&latitude=" + String(lat, 6) +
      "&longitude=" + String(lon, 6);

    http.begin(client, url);
    int code = http.GET();
    http.end();

    Serial.println(code == 200 ? "📍 Lokasi terkirim" : "❌ Gagal kirim lokasi!");
  }
}

void kirimTeksLokasi(float lat, float lon, float speed) {
  if (WiFi.status() == WL_CONNECTED) {
    WiFiClientSecure client;
    client.setInsecure();
    HTTPClient http;

    String pesan = "⚠️ Kecepatan melebihi batas!\n";
    pesan += "\xF0\x9F\x9A\x97 Kecepatan: " + String(speed, 1) + " km/jam";
    pesan += "\xF0\x9F\x93\x8D https://www.google.com/maps?q=" +
      String(lat, 6) + "," + String(lon, 6) + "\n";
  }
}

```

```

String url = "https://api.telegram.org/bot" + botToken +
  "/sendMessage?chat_id=" + chatID +
  "&text=" + urlencode(pesan);

http.begin(client, url);
int code = http.GET();
http.end();

Serial.println(code == 200 ? "📧 Pesan terkirim" : "❌ Gagal kirim pesan");
}
}

// ===== Fungsi Bantuan =====

String urlencode(String str) {
  String encoded = "";
  char c;
  char code0, code1;
  for (int i = 0; i < str.length(); i++) {
    c = str.charAt(i);
    if (isalnum(c)) {
      encoded += c;
    } else {
      code0 = (c & 0xf) + '0';
      if ((c & 0xf) > 9) code0 = (c & 0xf) - 10 + 'A';
      code1 = ((c >> 4) & 0xf) + '0';
      if (((c >> 4) & 0xf) > 9) code1 = ((c >> 4) & 0xf) - 10 + 'A';
      encoded += '%' + code0;
      encoded += code1;
      encoded += '%';
    }
  }
  return encoded;
}

String pad2(int number) {
  return (number < 10 ? "0" : "") + String(number);
}

```

VI KESIMPULAN

1. Sistem ini mampu melakukan pemantauan kecepatan serta lokasi kendaraan secara real-time dengan menggunakan sensor GPS Neo-6M yang diintegrasikan dengan mikrokontroler ESP32.
2. Sistem secara otomatis memberikan peringatan berupa notifikasi suara melalui modul DFPlayer Mini yang terhubung dengan speaker, serta mengirimkan informasi mengenai kecepatan dan lokasi kendaraan melalui aplikasi Telegram.
3. Pengujian sistem yang dilakukan pada kendaraan roda dua (sepeda motor) menunjukkan bahwa perangkat mampu memberikan peringatan secara tepat waktu dan akurat ketika kendaraan melampaui batas kecepatan.
4. Pengiriman data kecepatan dan lokasi kendaraan ke perangkat pemantau dilakukan secara efisien melalui koneksi internet menggunakan ESP32. Data dikirimkan secara otomatis ke aplikasi Telegram sehingga memudahkan petugas atau pengelola armada untuk melakukan pemantauan dari jarak jauh.

DAFTAR PUSTAKA

Ahdan, S., & Redy Susanto, E. (2021). Implementasi dashboard smart energy untuk pengontrolan rumah pintar pada perangkat bergerak berbasis internet of things. *Jurnal Teknoinfo*, 15(1), 26.
<https://doi.org/10.33365/jti.v15i1.954>

Ariefni, D. F., & Legowo, M. zB. (2018). Penerapan Konsep Monitoring Dan Evaluasi Dalam Sistem Informasi Kegiatan Mahasiswa Di Perbanas Institute Jakarta. *Jurnal Teknik Informatika Dan Sistem Informasi*, 4, 2229–2443.
<https://doi.org/10.28932/jutisi.v4i3.876>

Bosar Panjaitan, & Irfan Aprian Iswandana. (2023). Sistem Monitoring Ketinggian Dan Kecepatan Banjir Melalui Web Dan Peringatan Dini Melalui Telegram Berbasis Node Mcu. *Jurnal Limits*, 18(01), 8–18.
<https://doi.org/10.59134/jlmt.v18i01.201>

Darmawan, C. W., Sompie, S. R. U. A., & Kambey, F. D. (2020). Implementasi Internet of Things Pada Monitoring Kecepatan Kendaraan Bermotor. *Jurnal Teknik Elektro Dan Komputer*, 9(2).
<https://doi.org/10.35793/jtek.v9i2.29414>

Erawati, W. (2019). Perancangan Sistem Informasi Penjualan Dengan Pendekatan Metode Waterfall. *Jurnal Media Informatika Budidarma*, 3(1), 1.
<https://doi.org/10.30865/mib.v3i1.987>

Fitria Haya, R., Rizka Gunawan, C., & Amir, F. (2020). Monitoring System For Decorative Plants Using Arduino Nano Microcontroller. *ULTIMA Computing*, XII(2), 65.