

Research Article

Traffic Condition Classification Using IoT on Raden Inten II Road

Untung Surapati ¹, Yuma Akbar ², Dwi Swasono Rachmad ³, Hadi Gunawan ^{4,*}

1 Sekolah Tinggi Ilmu Komputer Cipta Karya Informatika (Stikomcki) Jakarta

2 Sekolah Tinggi Ilmu Komputer Cipta Karya Informatika (Stikomcki) Jakarta

3 Sekolah Tinggi Ilmu Komputer Cipta Karya Informatika (Stikomcki) Jakarta

4 Sekolah Tinggi Ilmu Komputer Cipta Karya Informatika (Stikomcki) Jakarta

* Corresponding Author: e-mail: hadig488@gmail.com

Abstract: Unmonitored traffic conditions often hinder decision-making processes in traffic management, particularly on secondary roads. Jalan Raden Inten II in East Jakarta is one of the connecting routes with heavy traffic activity at certain times, yet no integrated data-based monitoring system is currently available. This study proposes an Internet of Things (IoT)-based traffic condition classification system to identify Clear, Normal, or Congested states based on vehicle counts and speed categorization. The system is designed using an ESP32 microcontroller, an HB100 sensor to detect vehicle speed, and two AJ-SR04M ultrasonic sensors to detect vehicle presence. Data on vehicle counts and the percentage of slow-moving vehicles are periodically transmitted to the ThingSpeak platform and processed using the Threshold-Based Classification method. The classification results are visualized on a dashboard-based website equipped with charts, traffic condition status, and notifications when consecutive congestion is detected. Testing was conducted using simulation data over a specific period. Qualitative validation was carried out by comparing the classification results with traffic indicators from Google Maps. The results show that the system can classify traffic conditions with a good degree of agreement with external references, although discrepancies occurred at certain times due to the limitations of simulated data. This research demonstrates that a simple IoT approach can provide an affordable and effective solution for monitoring and classifying traffic conditions, with potential for real-world implementation in future studies.

Keywords: Internet of Things (IoT), Traffic Condition Classification, Traffic Monitoring System, Threshold-Based Classification, ThingSpeak.

Received: August 15, 2025

Revised: September 10, 2025

Accepted: September 25, 2025

Published: October 16, 2025

Curr. Ver.: October 28, 2025



Copyright: © 2025 by the authors.

Submitted for possible open

access publication under the

terms and conditions of the

Creative Commons Attribution

(CC BY SA) license

(<https://creativecommons.org/licenses/by-sa/4.0/>)

1. Introduction

Urban traffic is an important social and technological issue because it directly affects mobility, public productivity, travel efficiency, and local decision-making. In large cities such as Jakarta, traffic conditions change rapidly due to increasing vehicle volume and daily community activities, creating conditions that range from smooth movement to severe congestion. According to [1], low-cost sensing technologies can support road traffic monitoring by providing measurable data that are useful for understanding traffic patterns. However, traffic data are often more available on major roads than on secondary or neighborhood roads. This situation creates a gap in local traffic information, especially in areas where traffic conditions are not systematically recorded. Jalan Raden Inten II in East Jakarta represents this problem because it functions as a connecting road for commercial and residential activities but lacks a data-based monitoring system. Therefore, real-time and historical traffic documentation is needed to support adaptive local traffic management [2], [3].

Several previous studies have shown that sensor-based and IoT-based systems can be applied to vehicle detection, traffic light control, and transportation monitoring. According

to [4], ultrasonic and magnetic field sensors can detect vehicle presence and transmit detection results through a communication network. [5] also demonstrated that ultrasonic sensors can be used to support traffic light regulation by measuring vehicle distance. In addition, [6] developed an IoT-based radar visualization system using ESP32 and sensors to detect moving and stationary objects. Although these studies prove the potential of simple sensors in transportation systems, most existing works focus on detection, parking monitoring, or traffic light control rather than time-based traffic condition classification on secondary roads. This study offers novelty by combining an ESP32, HB100 speed sensor, and two AJ-SR04M ultrasonic sensors to classify traffic conditions based on vehicle count and speed categories [7], [8].

This study aims to develop an Internet of Things-based traffic condition classification system that can detect vehicle counts, identify vehicle speed categories, transmit data to a cloud platform, and classify traffic conditions automatically. According to [9], low-cost and noninvasive sensing technologies have strong potential to support practical traffic monitoring without relying on expensive infrastructure. Based on this perspective, the proposed system is designed to answer three main research questions. First, how can a simple and affordable IoT-based system be implemented to monitor traffic density on Jalan Raden Inten II, East Jakarta? Second, how can the system transmit traffic data in real time to support more accurate monitoring? Third, how can the collected data be analyzed to classify traffic conditions automatically into clear, normal, or congested categories? These questions guide the development of a threshold-based classification approach using data transmitted through ThingSpeak and visualized through a web-based dashboard [10], [11].

The main argument of this study is that a simple IoT system can provide practical, affordable, and understandable traffic information for local stakeholders who do not have access to advanced traffic monitoring infrastructure. According to [12], wireless and radio-based sensing approaches show that traffic monitoring does not always require expensive camera-based systems. In line with this view, this research contributes by presenting a low-cost prototype that focuses on hourly traffic classification rather than only detecting current traffic status. The system also contributes methodologically by using simulated data and qualitative validation through Google Maps traffic indicators, allowing the prototype to be evaluated without direct field deployment. Practically, the results can support local administrators in identifying busy hours, planning traffic-related recommendations, and communicating traffic information to residents through a simple dashboard. Therefore, this study contributes to the development of accessible smart-city solutions for secondary roads and densely populated urban areas [13], [14].

2. Literature Review

Systematic Literature Review (SLR) is a structured research method used to identify, evaluate, and synthesize previous studies that are relevant to a specific research topic. According to [1], a systematic review of low-cost sensing technologies is important because it helps researchers understand the strengths, limitations, and implementation opportunities of various traffic monitoring approaches. In this study, SLR is used to examine previous research related to IoT-based vehicle detection systems, sensor utilization, traffic monitoring, and classification methods. The SLR process consists of three main stages: planning, conducting, and reporting. The planning stage includes formulating research questions, determining inclusion and exclusion criteria, and preparing the literature search protocol. The conducting stage involves searching, selecting, evaluating, and extracting data from relevant publications. The reporting stage presents the findings, research gaps, and recommendations for further study. The PICOC approach is also applied to define population, intervention, comparison, outcome, and context in a systematic manner [7], [8].

Table 1. Review PICOC.

Title of Survey Theme :	
"Classification of Traffic Rush Hours using <i>IoT</i> in Duren Sawit District"	
Population	Classification and detection of the number of vehicles using IOT
Intervention	<ol style="list-style-type: none"> a. How to design and implement a simple and cost-effective IoT-based traffic detection system to monitor traffic conditions on Jalan Raden Inten II? b. How can the designed IoT system transmit real-time traffic data to support more accurate vehicle intensity monitoring?

	c. How traffic data is collected from IoT systems can be analyzed and used to classify traffic conditions in a timely manner automated and accurate in the research area?
Comparison	N/A
Outcomes	Detect the number of vehicles on the road well, and Can perform classification based on vehicle number detection
Context	Dataset private

Review Survei Protocol

Protocol Survey Review is the process of conducting a survey methodology in the collection of scientific journals in accordance with the thesis title and written in PINOC in point 1. The review of the Survey Protocol can be seen in table 2. below:

Table 2. Review Survei Protocol.

Review Survei Protocol	
Publication Year	2021 s/d 2025
Publication Type	<input type="checkbox"/>) <i>Journal</i> () <i>Conference</i> () <i>Proceeding</i> <input type="checkbox"/>) <i>Book Chapter</i>
Search String	("All Metadata" count detection using IoT) or ("All Metadata" on IoT implementation)
Final Selected	29

It is necessary to have stages in the search for research sources before it, to get research journals that have been published for it in the Study Selection Strategy or Studies Selection Strategy, which is to carry out the following stages, namely:

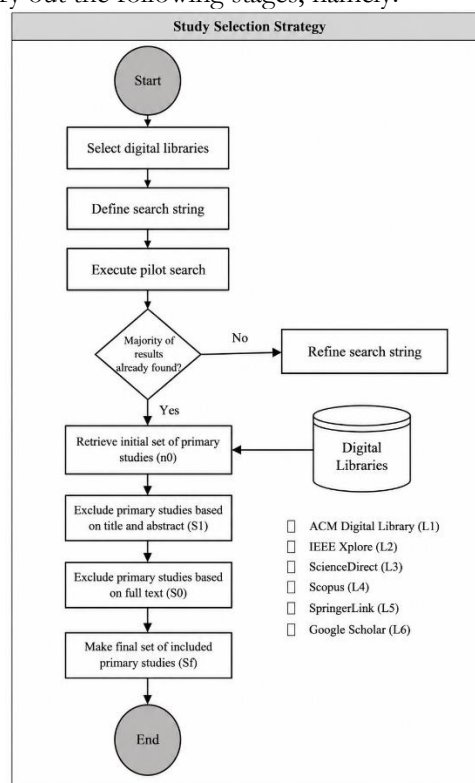


Figure 1. Studies Selection Strategy.

Conceptual Review

Conceptual review is an important part of a literature review because it explains the main concepts that support the research variables, system design, and analytical approach. According to [1], low-cost sensing technologies for road traffic monitoring need to be reviewed conceptually because each sensor has different characteristics, limitations, and implementation requirements. In this study, the conceptual review focuses on classification, traffic conditions, Internet of Things, ESP32, HB100 sensor, AJ-SR04M ultrasonic sensor,

ThingSpeak, and real-time monitoring. These concepts are interconnected because the proposed system depends on sensor-based vehicle detection, internet-based data transmission, and threshold-based data interpretation. A strong conceptual foundation helps clarify how vehicle count and speed category can be transformed into meaningful traffic status information. Therefore, the literature review does not only describe the hardware components but also explains how the system supports automated traffic condition classification in a local road environment [3], [9].

Classification

Classification is a systematic process of grouping data, objects, or events into certain categories based on relevant characteristics, values, or patterns. According to [8], traffic condition detection can be categorized into several states, such as quiet, normal, and congested, by using sensor data processed through an IoT-based system. In this study, classification is used to group traffic conditions into three categories: clear, normal, and congested. The classification process is based on vehicle count and the percentage of slow-moving vehicles detected within a certain time interval. The method applied is threshold-based classification, which determines a category by comparing input values with predefined numerical limits. This approach is suitable for simple IoT prototypes because it is easy to implement, does not require complex computation, and can run efficiently on embedded systems such as ESP32. Therefore, classification becomes the main decision-making mechanism in transforming raw sensor data into interpretable traffic information [5], [14].

Traffic Conditions

Traffic condition refers to the operational state of vehicle movement on a road segment during a particular time period. According to [7], vehicle detection using ultrasonic sensor networks can support the analysis of traffic flow by identifying the presence and movement of vehicles in monitored areas. In general, traffic conditions can be described as clear, normal, or congested depending on vehicle volume, average speed, road density, and travel delay. A clear condition occurs when vehicles move freely with minimal obstacles, while a normal condition indicates moderate movement with slight slowing. Congested traffic occurs when vehicle accumulation causes significant speed reduction or long queues. In this research, traffic condition is defined based on two measurable indicators: vehicle count and speed category. These indicators are collected through IoT sensors and interpreted using a threshold-based classification method. This definition allows traffic status to be analyzed periodically and supports local decision-making in secondary urban roads [1], [2].

Internet of Things (IoT)

Internet of Things is a technological concept in which physical devices, sensors, microcontrollers, and applications are connected through the internet to collect, transmit, process, and visualize data automatically. According to [10], IoT enables physical objects to communicate and exchange data through internet-based networks, making it useful for monitoring and automation systems. In transportation, IoT can support traffic monitoring, smart parking, congestion detection, adaptive traffic lights, and environmental sensing. An IoT system generally consists of sensor devices, communication networks, cloud platforms, and user interfaces. In this study, IoT is applied to connect vehicle detection sensors with ESP32 and ThingSpeak so that traffic data can be transmitted periodically. The use of IoT makes the system more practical because data can be accessed remotely and visualized through a dashboard. This concept supports affordable and scalable traffic monitoring, especially for local roads that do not yet have integrated monitoring infrastructure [3], [15].

ESP32



Figure 2. ESP32.

ESP32 is a microcontroller-based System-on-Chip widely used in Internet of Things projects because it provides built-in Wi-Fi, Bluetooth, sufficient processing capability, and compatibility with various sensor interfaces. According to [11], ESP32 is suitable for IoT development because it supports wireless communication, digital and analog input, and efficient data processing for sensor-based applications. Compared with earlier modules such as ESP8266, ESP32 offers improved performance through a dual-core processor, more GPIO pins, and better flexibility for real-time monitoring systems. In this research, ESP32 functions as the main controller that receives signals from the HB100 sensor and AJ-SR04M ultrasonic sensors, processes the input data, and sends the results to ThingSpeak. Its support for Wi-Fi communication allows vehicle detection data to be transmitted without additional network modules. However, analog signal processing still requires careful calibration because ADC readings may be affected by noise, especially when wireless communication is active [16], [17].

HB100 Sensor



Figure 3. HB100 Sensor.

The HB100 sensor is a microwave motion sensor based on the Doppler effect, commonly used to detect moving objects and estimate relative speed. According to [12], non-camera sensing approaches can support vehicle detection and classification without depending on visual conditions, making them useful for low-cost traffic monitoring. The HB100 operates by transmitting microwave signals and receiving reflected waves from moving objects. When a vehicle moves in front of the sensor, the reflected frequency changes, producing an analog signal that can be processed by a microcontroller. In this study, the HB100 sensor is used to support vehicle speed categorization, particularly to distinguish between faster and slower vehicle movement. Its advantages include low cost, non-contact operation, fast response, and independence from lighting conditions. However, it cannot detect stationary objects and cannot count vehicles accurately by itself. Therefore, it needs to be combined with ultrasonic sensors to improve vehicle presence detection and traffic classification reliability [6], [9].

AJ-SR04M Ultrasonic Sensor



Figure 4. AJ-SR04M Ultrasonic Sensor.

The AJ-SR04M ultrasonic sensor is a waterproof distance sensor designed for outdoor applications, including vehicle presence detection, parking systems, and environmental monitoring. According to [13], ultrasonic sensors can provide accurate distance measurement and are effective for detecting parking slot availability when integrated with microcontroller-based systems. The AJ-SR04M works by emitting ultrasonic waves and measuring the time required for the reflected signal to return from an object. This mechanism allows the sensor to estimate object distance and detect whether a vehicle is present within a certain range. In this research, two AJ-SR04M sensors are used to support vehicle detection and counting. Their waterproof design makes them suitable for outdoor road environments, while their compatibility with ESP32 allows simple integration through trigger–echo or serial communication modes. However, ultrasonic sensors may be affected by object angle, environmental noise, and placement position. Therefore, proper calibration and positioning are needed to produce reliable traffic monitoring data [4], [7].

ThingSpeak



Figure 5. ThingSpeak.

ThingSpeak is a cloud-based IoT platform used to collect, store, analyze, and visualize sensor data from internet-connected devices. According to [8], ThingSpeak can be used in IoT-based traffic congestion detection systems to receive real-time sensor data and display monitoring results through online visualization. The platform supports data transmission through REST API, channel-based storage, graphical visualization, and MATLAB-based analysis. In this study, ThingSpeak is used as the cloud platform for storing vehicle count and speed category data sent by ESP32. The collected data can then be analyzed to classify traffic conditions into clear, normal, or congested categories. ThingSpeak is suitable for academic prototypes because it is easy to configure, supports real-time charts, and does not require researchers to build an independent server. However, its free version has limitations in update intervals, channel capacity, and real-time response speed. Even so, it remains practical for low-cost IoT traffic monitoring systems [18], [19].

Real-Time Monitoring

Real-time monitoring is the process of observing, collecting, transmitting, and analyzing data as events occur or within a short time delay. According to [2], autonomous wireless sensor systems are useful for road monitoring because they can provide timely information in areas with limited communication coverage. In IoT-based traffic systems, real-time monitoring allows users to view current traffic conditions, detect congestion patterns, and respond quickly to changes in vehicle flow. This research applies real-time monitoring by sending sensor data from ESP32 to ThingSpeak periodically and displaying the classification results through a web-based dashboard. The system enables traffic status to be observed without manual counting or direct field observation. Real-time monitoring is especially relevant for local roads because traffic conditions can change quickly during morning and evening peak hours. By combining sensor detection, cloud storage, and dashboard visualization, the system provides practical traffic information for local administrators and community stakeholders [14], [15].

3. Materials and Method

Research Data

The data used in this study are classification data designed to group traffic conditions into three categories: Clear, Normal, and Congested, based on the number of vehicles detected per hour. This study applies a simulated experimental approach, in which the system is tested using dummy data to produce logical traffic condition classifications without direct field testing. The data were collected for seven days, from June 7 to June 13, 2025, during the daily period from 05:00 to 21:00 WIB, which is considered representative of traffic operating hours in a dense urban area such as Jalan Raden Inten II, East Jakarta. The collected data include detection time, hourly vehicle count, number of fast-moving vehicles, and number of slow-moving vehicles. These data are transmitted in real time to the ThingSpeak platform, so manual downloading in file formats such as CSV is not required. Since the dataset does not come from public sources or institutions, it is categorized as private research data and is used only within the scope of this study. Dataset validation is conducted qualitatively by comparing the classification results with Google Maps traffic color indicators at corresponding times.

Methodology Implementation

The methodology implementation began with the data collection process using a simple Internet of Things (IoT) system consisting of an ESP32 microcontroller, one HB100 microwave sensor, and two AJ-SR04M ultrasonic sensors. The system was designed to detect vehicle presence, estimate vehicle speed categories, and transmit data automatically to the

ThingSpeak cloud platform. Data collection was conducted in a simulated environment representing traffic conditions on Jalan Raden Inten II, East Jakarta, from June 7 to June 13, 2025, between 05:00 and 21:00 WIB. The detection process started by reading analog signals from the HB100 sensor to identify significant movement. When a predefined signal change was detected, the ultrasonic sensors were activated to measure the distance of the detected object. Vehicles located within a distance range of 1–4 meters were further analyzed based on distance stability over a minimum period of three seconds. Stable measurements were categorized as slow-moving vehicles, while unstable measurements were categorized as normal or fast-moving vehicles. After each detection, the system entered a cooldown phase to prevent duplicate counting and subsequently transmitted the detection time, vehicle category, and vehicle count to ThingSpeak through a Wi-Fi connection.

After the data had been collected and stored in ThingSpeak, a preprocessing stage was performed to aggregate vehicle records into hourly summaries consisting of total vehicle counts, fast-moving vehicles, and slow-moving vehicles. These processed data served as inputs for the traffic condition classification stage. The study employed a Threshold-Based Classification method that categorized traffic conditions into Clear, Normal, or Congested based on two main parameters: the total number of vehicles detected per hour and the percentage of slow-moving vehicles. The classification results were then utilized in a web-based dashboard to provide visual representations of traffic conditions and support decision-making. In addition, an automatic notification mechanism was implemented to identify recurring congestion events. Notifications were generated when the system detected a “Congested” status for three consecutive hours, providing an early warning for local traffic administrators. This visualization and notification framework ensured that the collected data were not only stored and analyzed but also transformed into actionable information capable of supporting real-time traffic monitoring and local traffic management strategies.

System Testing Design

The system testing phase was designed to evaluate the functionality and reliability of the proposed IoT-based vehicle detection system. This testing aimed to ensure that all hardware components, including the ESP32 microcontroller, the HB100 microwave sensor, and the two AJ-SR04M ultrasonic sensors, operated according to the designed system logic and were capable of transmitting data automatically to the ThingSpeak cloud platform. The evaluation was conducted using a simulated real-time approach in which several vehicle movement scenarios were represented, including fast-moving, slow-moving, and stationary objects positioned in front of the sensors. During the testing process, the system's ability to detect vehicle presence, classify movement characteristics, count detected vehicles, and transmit the resulting data to the cloud platform was observed and verified. The testing also examined whether the collected data could be stored correctly and subsequently used for traffic condition classification. Through this testing procedure, the overall performance of the IoT system was assessed to ensure that it could provide accurate, consistent, and reliable input data for the subsequent traffic monitoring and classification processes.

Data Accuracy and Classification Feasibility Testing

This testing phase was conducted to evaluate whether the vehicle data collected and transmitted by the IoT system to the ThingSpeak platform could be used reliably for traffic condition classification. The testing process began with system initialization, including establishing a Wi-Fi connection for the ESP32, configuring communication with ThingSpeak, calibrating the HB100 sensor using a baseline ADC value of 3100, and setting the AJ-SR04M ultrasonic sensor detection range between 1 and 4 meters. During operation, the system continuously monitored the HB100 sensor to identify significant ADC changes indicating vehicle movement. Vehicles exhibiting rapid movement were detected when the ADC variation exceeded the predefined threshold, while slow-moving vehicles were identified through ultrasonic distance measurements and stability analysis. A cooldown mechanism was implemented to prevent duplicate detections by temporarily suspending new detections until the detected object left the sensing area or experienced a significant distance change. Vehicle data were transmitted to ThingSpeak every 60 seconds and aggregated on an hourly basis. For traffic classification, the system calculated the total number of detected vehicles (N_{total}) and the number of slow-moving vehicles (N_{slow}), then determined the percentage of slow-moving vehicles using the formula $(P_{slow} = \frac{N_{slow}}{N_{total}} \times 100\%)$. The resulting values were subsequently used as inputs for the threshold-based classification method to categorize traffic conditions as Clear, Normal, or Congested.

Perform a classification of traffic conditions based on the threshold table as follows:

Table 3. Traffic Classification Criteria.

Traffic Classification Criteria		
Number of Vehicles (per hour)	Percentage of Vehicles Slow	Condition Classification
< 5	< 50%	Smooth
< 5	\geq 50%	Solid
5 – 15	< 50%	Normal
5 – 15	\geq 50%	Solid
> 15	< 40%	Smooth
> 15	\geq 40%	Solid

This classification aims to capture traffic conditions that are not only based on the quantity of vehicles, but also the quality of their movement. For example, here is the result of simulation data for one of the hours (e.g. 07.00–08.00 WIB).

Table 4. Examples of Data Results and Classification.

Time	Number of Vehicles (N_total)	Slow Vehicles (N_lambat)	Slow Percentage (P_lambat)	Klasifikasi
07:00–08:00	18	10	55.56%	Solid
08:00–09:00	12	3	25.00%	Normal
09:00–10:00	4	1	25.00%	Smooth
10:00–11:00	17	5	29.41%	Smooth
11:00–12:00	6	4	66.67%	Solid

Note: The table above contains sample data generated from the processing of the researcher's ThingSpeak channel during the simulation period conducted from June 7 to June 13, 2025. The testing process was considered successful if the system consistently recorded the number of detected vehicles and accurately differentiated between fast-moving and slow-moving vehicles. In addition, the percentage of slow-moving vehicles had to be calculated correctly based on the collected data, and the resulting traffic classifications had to remain consistent with the predefined threshold-based classification rules. These indicators were used to evaluate the reliability of the system and to verify that the collected data were suitable for traffic condition analysis and automated classification.

Pengujian Visualisasi dan Notifikasi Informasi Lalu Lintas

Pengujian ini bertujuan untuk mengevaluasi kelayakan tampilan website sebagai media penyampaian informasi hasil klasifikasi kondisi lalu lintas kepada pihak non-teknis, khususnya pengurus wilayah di sekitar Jalan Raden Inten II. Website dikembangkan sebagai dashboard internal yang menampilkan hasil pengolahan data dari sistem IoT secara ringkas, interaktif, dan mudah dipahami. Beberapa elemen visual yang ditampilkan dan diuji pada website dapat dilihat pada Tabel

5 Visualization Testing below.

Table 5. Visualization Testing.

No.	Visualization Elements	Description
1	Graph of the number of vehicles per hour	Bar or line graph to show the number of vehicles every hour from 05:00–21:00
2	Traffic hour classification	Color tables or indicators (red, yellow, green) based on hourly classification results
3	Percentage of slow vehicles vs fast	Pie charts or tables showing Composition of vehicle movement
4	Congested condition notifications consecutive	Automatic alert display if the system detects \geq 3 solid hours in a row
5	Recommended intervention time	Specific hours recommendations for traffic settings based on classification data
6	Daily data logs	Downloadable hourly and per-day data tables in CSV or PDF format
7	Weekly reports	Button to download the recap report classification results in PDF format

Visualization and Accuracy Testing

Visualization and accuracy testing were conducted to evaluate both the usability of the web-based dashboard and the reliability of the IoT vehicle detection system. The visualization testing aimed to ensure that all processed data from the IoT system were correctly displayed on the website through graphs, tables, traffic condition indicators, and automatic notifications. The dashboard was designed to be easily understood by both technical and non-technical users, enabling local administrators to interpret traffic conditions without requiring specialized knowledge. The testing was considered successful if all classification data retrieved from ThingSpeak were displayed accurately, notification alerts for consecutive congestion events appeared automatically, and users could access or download traffic reports for documentation purposes. In addition, accuracy testing was performed directly on Jalan Raden Inten II by comparing the number of vehicles detected by the IoT system with the actual number of vehicles counted manually through direct observation. The device was positioned beside the road and operated for a five-minute observation period. The detected vehicle count obtained from ThingSpeak was then compared with the manually recorded count, and the system accuracy was calculated using the formula: $\text{Accuracy (\%)} = (\text{Detected Vehicles} / \text{Actual Vehicles}) \times 100\%$.

4. Results and Discussion

Research Instruments

This study utilized a combination of hardware and software components to develop an Internet of Things (IoT)-based traffic condition classification system. The hardware configuration consisted of an ESP32 development board as the main controller, an HB100 microwave radar sensor for detecting vehicle movement, and an AJ-SR04M waterproof ultrasonic sensor for identifying vehicle presence and supporting speed categorization. Additional components included a TP4056 charging module with battery protection, two 18650 Li-ion batteries with a battery holder, a 5V step-up boost converter, and a breadboard with jumper wires for circuit assembly and testing. A laptop or computer was used for system programming, monitoring, and data analysis. On the software side, Arduino IDE was employed to develop and upload the ESP32 firmware, while ThingSpeak served as the cloud platform for real-time data storage and monitoring. Google Chrome was used to access the IoT dashboard and cloud services, whereas Google Maps provided visual traffic references for qualitative validation. Furthermore, a private web-based dashboard developed using HTML, CSS, JavaScript (Chart.js), Bootstrap, and XAMPP was used to visualize traffic conditions, display notifications, and generate traffic monitoring reports.

Implementation and Testing

The traffic condition classification system developed in this study was implemented using an Internet of Things (IoT) architecture centered on an ESP32 microcontroller. The system was designed to automatically detect vehicle presence, estimate movement characteristics, and transmit traffic data to the ThingSpeak cloud platform through a Wi-Fi connection. The ESP32 served as the primary processing unit, responsible for sensor data acquisition, local data processing, and communication with the cloud server. Vehicle movement detection was performed using an HB100 microwave Doppler radar sensor, which identifies moving vehicles by measuring frequency shifts in reflected signals, enabling basic speed estimation. To complement this functionality, an AJ-SR04M waterproof ultrasonic sensor was employed to detect vehicle presence and measure distance, supporting the classification of vehicle movement patterns. The system was powered by a portable energy module consisting of two 18650 Li-ion batteries and a TP4056 charging module with overcharge and over-discharge protection. A 5V step-up boost converter was used to stabilize the power supply required by the ESP32 and sensing components, ensuring reliable operation during continuous traffic monitoring activities.



Figure 6. Physical Display of Vehicle Detection Device.

The implemented IoT system operates through a sequence of automated processes designed to detect vehicles, classify their movement characteristics, and transmit traffic information to a cloud platform. Initially, the ESP32 establishes connections with the Wi-Fi network, ThingSpeak server, and all sensing modules. The HB100 Doppler radar sensor continuously monitors vehicle movement and identifies fast-moving vehicles when signal variations exceed a predefined threshold. Vehicle presence is then validated using the ultrasonic sensor, where objects detected within a specified distance range are classified as fast-moving vehicles. When the HB100 sensor is not triggered, the system relies on ultrasonic distance variations to identify slow-moving vehicles. Vehicles remaining stationary for more than five seconds are interpreted as indicators of congestion and are recorded separately for traffic condition analysis rather than being counted as moving vehicles. Traffic data, including vehicle counts and movement categories, are accumulated and transmitted automatically to the ThingSpeak cloud platform at intervals of at least 60 seconds. After each transmission cycle, the counters are reset, and monitoring resumes. The entire system is programmed using Arduino IDE, which manages sensor initialization, data acquisition, vehicle classification, and cloud communication through Wi-Fi connectivity.

Kode:

```
#include <WiFi.h>
#include "ThingSpeak.h"

// === Konfigurasi WiFi & ThingSpeak ===
const char* ssid = "Punya";
const char* password = "hadi12345";

WiFiClient client;

unsigned long channelID = 3008261;
const char* writeAPIKey = "O7QC4ZPTU3KJ2HM0";

// === Pin Sensor ===
const int pinHB100_adc = 34; // ADC untuk HB100
const int trigPin = 25;
const int echoPin = 26;

// === Parameter Kalibrasi ===
const int HB_THRESHOLD = 100; // Perubahan ADC minimum untuk deteksi
const int HB_BASELINE = 3100; // Nilai baseline ADC saat idle
const int JARAK_MIN = 100; // 1 meter dalam cm
const int JARAK_MAX = 400; // 4 meter dalam cm

const unsigned long DEBOUNCE_MS = 300; // Minimal interval antar deteksi
const unsigned long MACET_TIME = 3000; // Waktu stabil untuk deteksi macet
const unsigned long SEND_INTERVAL = 60000; // Interval kirim data ke ThingSpeak

// === Variabel Sistem ===
int count_cepat = 0;
```

```
int count_lambat = 0;

unsigned long lastDetectTime = 0;
unsigned long lastSendTime = 0;
unsigned long distanceStableTime = 0;

bool vehiclePresent = false;
bool macetStatus = false;

float lastDistance = 0;

// === Setup Awal ===
void setup() {
  Serial.begin(115200);

  pinMode(pinHB100_adc, INPUT);
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);

  // Koneksi WiFi
  WiFi.begin(ssid, password);

  Serial.print("Menghubungkan ke WiFi");
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }

  Serial.println("\nWiFi connected");

  // Inisialisasi ThingSpeak
  ThingSpeak.begin(client);
}

// === Fungsi Membaca Sensor Ultrasonik ===
float readUltrasonic() {
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);

  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);

  long duration = pulseIn(echoPin, HIGH, 30000); // Timeout 30 ms

  // Konversi durasi ke jarak cm
  float distance = (duration * 0.0343) / 2;

  return distance;
}

// === Fungsi Kirim Data ke ThingSpeak ===
void sendToThingSpeak() {
  ThingSpeak.setField(1, count_cepat); // Field 1: Kendaraan cepat
  ThingSpeak.setField(2, count_lambat); // Field 2: Kendaraan lambat
  ThingSpeak.setField(3, macetStatus); // Field 3: Status macet 0/1

  int statusCode = ThingSpeak.writeFields(channelID, writeAPIKey);

  Serial.print("Update ThingSpeak status: ");
  Serial.println(statusCode);
}
```

```

}

// === Loop Utama ===
void loop() {
  int hbValue = analogRead(pinHB100_adc);
  float distance = readUltrasonic();
  unsigned long currentTime = millis();

  // Debug output
  Serial.print("HB: ");
  Serial.print(hbValue);
  Serial.print(" | Jarak: ");
  Serial.print(distance);
  Serial.print(" cm | Cepat: ");
  Serial.print(count_cepat);
  Serial.print(" | Lambat: ");
  Serial.print(count_lambat);
  Serial.print(" | Macet: ");
  Serial.println(macetStatus);

  // Deteksi perubahan jarak signifikan
  if (abs(distance - lastDistance) > 10) {
    distanceStableTime = currentTime;
    lastDistance = distance;
  }

  // Jika objek berada dalam rentang deteksi
  if (distance >= JARAK_MIN && distance <= JARAK_MAX) {

    // Deteksi kendaraan cepat berdasarkan HB100
    if (abs(hbValue - HB_BASELINE) >= HB_THRESHOLD) {
      if (!vehiclePresent && (currentTime - lastDetectTime) > DEBOUNCE_MS) {
        count_cepat++;
        vehiclePresent = true;
        lastDetectTime = currentTime;

        Serial.println("Kendaraan cepat terdeteksi!");
      }
    }

    // Deteksi kendaraan lambat berdasarkan ultrasonik
    else if (!vehiclePresent) {
      if ((currentTime - distanceStableTime) < MACET_TIME) {
        if ((currentTime - lastDetectTime) > DEBOUNCE_MS) {
          count_lambat++;
          vehiclePresent = true;
          lastDetectTime = currentTime;

          Serial.println("Kendaraan lambat terdeteksi!");
        }
      }
    }

    // Deteksi kondisi macet
    if ((currentTime - distanceStableTime) >= MACET_TIME) {
      if (!macetStatus) {
        macetStatus = true;
        Serial.println("Kondisi macet terdeteksi!");
      }
    }
  } else {
    macetStatus = false;
  }
}

```

```

    }

    } else {
    vehiclePresent = false;
    macetStatus = false;
    }

    // Kirim data ke ThingSpeak setiap 60 detik
    if (currentTime - lastSendTime >= SEND_INTERVAL) {
    sendToThingSpeak();

    count_cepat = 0;
    count_lambat = 0;
    lastSendTime = currentTime;
    }

    delay(100);
}

```

Data visualization and traffic condition classification were implemented through a web-based dashboard designed to present vehicle detection results in an informative and user-friendly manner. The dashboard serves as the primary interface for monitoring traffic conditions by displaying processed data collected from the IoT system. The visualized information includes timestamps, total vehicle counts per hour, the number of slow-moving vehicles, the percentage of slow vehicles relative to the total traffic volume, and the resulting traffic classification categories, namely Smooth, Normal, and Congested. By presenting these indicators through graphical and tabular formats, the dashboard enables users to quickly understand traffic patterns and supports data-driven decision-making for local traffic management and monitoring activities.

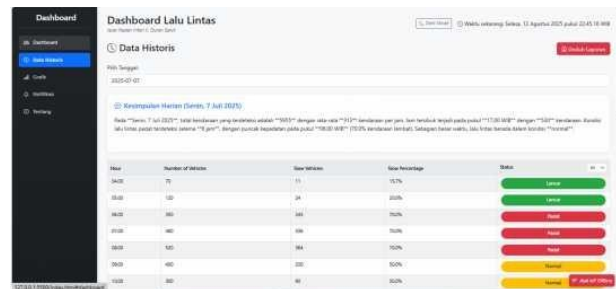


Figure 7. Traffic State Classification Results.

The classification is done based on Threshold-based classification, which compares the number of vehicles and the percentage of slow vehicles with a certain threshold.

// Traffic Condition Classification Logic

```

if (jumlah_kendaraan < 5 && persen_lambat < 50) {
    status = "Lancar";
}
else if (jumlah_kendaraan < 5 && persen_lambat >= 50) {
    status = "Padat";
}
else if (jumlah_kendaraan <= 15 && persen_lambat < 50) {
    status = "Normal";
}
else if (jumlah_kendaraan <= 15 && persen_lambat >= 50) {
    status = "Padat";
}
else if (jumlah_kendaraan > 15 && persen_lambat >= 40) {
    status = "Padat";
}
}

```

```

else {
    status = "Lancar";
}

```

The traffic classification results were presented through an interactive dashboard using multiple visualization formats to facilitate data interpretation and decision-making. These visualizations included bar charts displaying the number of detected vehicles per hour, line charts illustrating daily traffic density trends over time, and an automated notification system that generated alerts whenever the traffic status was classified as “Congested” for several consecutive periods. This combination of graphical representations and real-time alerts enabled users to quickly identify traffic patterns, monitor congestion levels, and respond proactively to recurring traffic problems in the monitored area.



Figure 8. Traffic Status Dashboard.

The simulation testing phase was conducted to verify that the proposed traffic classification logic operated correctly according to the predefined threshold-based classification method without requiring direct field deployment. The simulation utilized vehicle detection data collected and transmitted by the IoT system to the ThingSpeak platform during the observation period from 7–13 June 2025 between 05:00 and 21:00 WIB. The dataset included hourly vehicle counts, the number of slow-moving vehicles, the percentage of slow vehicles, and the resulting traffic classifications categorized as Smooth, Normal, or Congested. The testing process evaluated whether the classification rules produced consistent and logical outputs and whether the dashboard accurately visualized the processed data. The web-based dashboard retrieved data from ThingSpeak through an HTTP API and displayed the results using bar charts, traffic status indicators, historical logs, and daily trend summaries. In addition, an automatic notification mechanism continuously monitored traffic conditions and generated warning messages whenever the “Congested” status appeared consecutively for two or more periods within a short time interval. These notifications were presented through dashboard alerts and highlighted visual indicators to support timely decision-making by local traffic administrators.

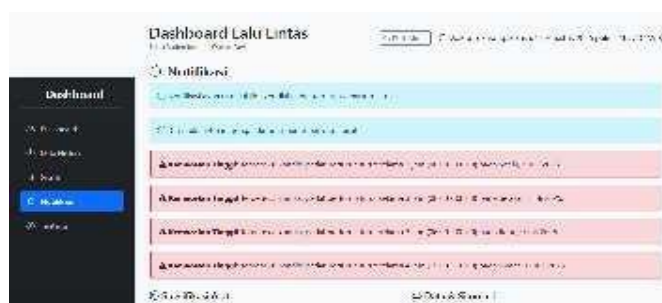


Figure 9. Notifications Page.

The evaluation results indicate that the dashboard visualization and notification features were successfully implemented and functioned as intended. The visual representations effectively helped users understand daily traffic patterns and identify periods of increased congestion. The automated congestion notifications proved useful as an early warning mechanism, particularly during peak traffic hours, enabling faster awareness and response. Although the system performed reliably, further improvements are recommended to optimize automatic data refresh rates and notification response times, especially under unstable or slow network conditions. Overall, the dashboard provided an intuitive and user-friendly interface, while the notification system supported operational decision-making processes, such as traffic signal timing adjustments and community traffic management initiatives.

Final Testing Results

The implementation results demonstrate that the proposed IoT-based traffic classification system successfully integrated hardware and software components into a functional monitoring platform. The system utilized an ESP32 microcontroller in combination with an HB100 Doppler radar sensor and AJ-SR04M ultrasonic sensors to detect vehicle movement, estimate speed categories, and record vehicle counts automatically. During testing, the ultrasonic sensors achieved reliable object detection within an effective range of approximately 1–4 meters, while the counting mechanism consistently recorded vehicle movements without duplication. Vehicle categories were successfully differentiated into fast-moving and slow-moving groups according to the predefined detection logic. Furthermore, all collected data, including vehicle counts, speed categories, and detection timestamps, were transmitted periodically to the ThingSpeak cloud platform without data loss, confirming the reliability of the communication and data acquisition processes. These results indicate that the system architecture functioned as intended and provides a solid foundation for future real-world deployment and field validation.

Simulation testing further verified the effectiveness of the proposed traffic classification framework. Vehicle detection data were transmitted automatically to ThingSpeak and processed using the Threshold-Based Classification method to categorize traffic conditions as Smooth, Normal, or Congested. The results showed that data transmission was consistently successful throughout the testing period, with no corrupted or missing records observed. The classification algorithm produced stable and logically consistent outputs across different traffic scenarios and observation periods. Changes in traffic status occurred only when vehicle counts and slow-vehicle percentages exceeded predefined threshold values, demonstrating the robustness of the classification rules. In addition, the generated datasets were sufficiently structured for dashboard visualization and qualitative validation using external traffic information sources. Overall, the simulation results confirmed that the proposed workflow, from data acquisition and cloud transmission to traffic classification, operated effectively and provided reliable outputs for subsequent visualization and analysis stages.

Table 6. Vehicle Detection Device Accuracy Test Results (5 Minutes Duration).

Recording Methods	Number of Vehicles
Manual Observation	134 Vehicles
IoT Tool Detection	122 Vehicles

From these results, a difference of 3 vehicles was obtained between the results of manual observation and the detection results of IoT devices. Using the accuracy formula:

$$A(0/0) = \frac{122}{134} \times 100\% = 91.04\% \quad \dots\dots\dots (1)$$

The website evaluation demonstrated that the dashboard successfully visualized traffic information and supported user-oriented decision making. The system displayed vehicle counts, slow-vehicle percentages, traffic classifications, and congestion notifications accurately based on data retrieved from ThingSpeak. Visualization components, including charts and status indicators, were responsive and easily accessible on both desktop and mobile devices. The automatic notification feature effectively alerted users whenever congestion conditions persisted for consecutive monitoring periods, providing valuable early-warning information. In addition, a field accuracy test conducted on Jalan Raden Inten II compared IoT-based vehicle detection with manual observation. During a five-minute observation period, the system detected 122 vehicles compared with 134 vehicles recorded manually, resulting in an accuracy rate of 91.04%. Although minor discrepancies occurred due to sensor positioning, closely spaced vehicles, and temporary vehicle stops, the accuracy level exceeded 90%, indicating that the proposed IoT system is sufficiently reliable for practical traffic monitoring and classification applications.



Figure 10. Vehicle Detection Device Accuracy Testing Documentation.

5. Conclusion

This study successfully designed, implemented, and evaluated an Internet of Things (IoT)-based traffic condition classification system for monitoring traffic density on Jalan Raden Inten II, East Jakarta. The proposed system integrated an ESP32 microcontroller, an HB100 Doppler radar sensor, and AJ-SR04M ultrasonic sensors to automatically detect vehicle movements, estimate vehicle speed categories, and record traffic data in real time. The collected data were transmitted to the ThingSpeak cloud platform, enabling continuous monitoring, storage, and subsequent analysis. The implementation results demonstrated that the system was capable of consistently detecting vehicle presence, distinguishing between fast-moving and slow-moving vehicles, and generating structured datasets suitable for traffic analysis and visualization.

The Threshold-Based Classification approach was successfully applied to categorize traffic conditions into Smooth, Normal, and Congested states based on the total number of detected vehicles and the percentage of slow-moving vehicles within a specified time interval. Simulation testing confirmed that the classification rules operated consistently across different traffic scenarios and observation periods. Furthermore, the dashboard application effectively visualized traffic information through charts, status indicators, and automated congestion notifications, making the results accessible and understandable for non-technical users. The integration between the IoT devices, cloud platform, and visualization dashboard functioned reliably throughout the testing process.

The accuracy evaluation conducted through a comparison between manual vehicle counting and IoT-based detection showed an accuracy rate of 91.04%, indicating that the proposed system is sufficiently reliable for practical traffic monitoring applications. Although minor discrepancies were observed due to sensor positioning and traffic dynamics, the overall performance demonstrates the feasibility of employing low-cost IoT technologies for traffic condition monitoring and classification. The findings suggest that the proposed approach can provide useful traffic information for local traffic management and decision-making while maintaining affordability and ease of deployment. Future work should focus on long-term field deployment, integration with additional traffic data sources, enhancement of classification models, and optimization of power management to improve system scalability, robustness, and real-world applicability.

References

- [1] M. Bernas, B. Placzek, W. Korski, P. Loska, J. Smyla, and P. Szymala, "A Survey and Comparison of Low-Cost Sensing Technologies for Road Traffic Monitoring," *Sensors*, vol. 18, no. 10, pp. 1–27, 2018, doi: 10.3390/s18103243.
- [2] M. Postigo-Malaga, A. M. Jimenez-Caceres, J. Pelegri-Sebastia, and J. Chilo, "Autonomous Wireless Sensor System for Emergency Monitoring Roads with Low Communication Coverage," *Electronics*, vol. 12, no. 23, 2023, doi: 10.3390/electronics12234829.
- [3] F. Akhter, S. Khadivizand, H. R. Siddiquei, M. E. E. Alahi, and S. Mukhopadhyay, "IoT Enabled Intelligent Sensor Node for Smart City: Pedestrian Counting and Ambient Monitoring," *Sensors*, vol. 19, no. 15, 2019, doi: 10.3390/s19153374.
- [4] F. Afrizal and B. N. Prastowo, "Vehicle Detection System Using Ultrasonic And Magnetic Fields Sensors Based On LoRa Communication," *Indones. J. Electron. Instrum. Syst.*, vol. 12, no. 2, pp. 157–168, 2022, doi: 10.22146/ijeis.71725.

- [5] D. O. Deltania, Djuniadi, and E. Apriaskar, "Pengaturan Lampu Lalu Lintas (Traffic Light) dengan Sensor Ultrasonik," *JETRI J. Ilm. Tek. Elektro*, vol. 19, no. 1, pp. 77–95, 2021, doi: 10.25105/jetri.v19i1.8660.
- [6] D. K. Ihsan, U. Fadlilah, and K. Muhammad, "Design and Development of Object Detection Radar with IoT-Based Matlab Software Visualization," *EMITOR*, pp. 154–160, 2024, doi: 10.23917/emitor.v24i2.5535.
- [7] Y. Jo and I. Jung, "Analysis of Vehicle Detection with WSN-Based Ultrasonic Sensors," *Sensors*, vol. 14, no. 8, pp. 14050–14069, 2014, doi: 10.3390/s140814050.
- [8] P. A. Rosyady, M. R. Feter, and Z. A. Ikhsan, "Prototipe Sistem Deteksi Kemacetan Jalan Raya Berbasis Internet of Things (IoT)," *Avitec*, vol. 4, no. 2, pp. 197–206, 2022, doi: 10.28989/avitec.v4i2.1270.
- [9] K. Wang, H. Xiong, J. Zhang, H. Chen, D. Dou, and C. Z. Xu, "SenseMag: Enabling Low-Cost Traffic Monitoring Using Noninvasive Magnetic Sensing," *IEEE Internet Things J.*, vol. 8, no. 22, pp. 16666–16679, 2021, doi: 10.1109/JIOT.2021.3074907.
- [10] L. P. I. Kharisma, Kelvin, and others, *Internet of Things: Pengenalan dan Penerapan Teknologi IoT*. PT Sonpedia Publishing Indonesia, 2024.
- [11] D. E. Putra, E. Rosman, H. Amnur, K. G. Flomina, M. Hasanah, and R. I. Salam, *Konsep Dasar Internet of Things (IoT) dengan Mikrokontroler ESP32*. Pustaka Galeri Mandiri, 2025.
- [12] B. Sliwa, N. Piatkowski, and C. Wietfeld, "The Channel as a Traffic Sensor: Vehicle Detection and Classification Based on Radio Fingerprinting," *IEEE Internet Things J.*, vol. 7, no. 8, pp. 7392–7406, 2020, doi: 10.1109/JIOT.2020.2983207.
- [13] K. Auliya, M. Yusfi, and R. Rasyid, "Sistem Pemantauan Slot Parkir Menggunakan Sensor Ultrasonik JSN-SR04T dan Pengenalan Plat Nomor Kendaraan dengan ESP32-CAM," *J. Fis. Unand*, vol. 12, no. 4, pp. 534–540, 2023, doi: 10.25077/jfu.12.4.534-540.2023.
- [14] T. Saputra and U. Surapati, "Analysis of the Effectiveness of IoT-Based Automatic Street Lighting Control Using Linear Regression Method," *Int. J. Softw. Eng. Comput. Sci.*, vol. 4, no. 2, pp. 690–701, 2024, doi: 10.35870/ijsecs.v4i2.2878.
- [15] L. M. Pires, J. Figueiredo, R. Martins, and J. Martins, "IoT-Enabled Real-Time Monitoring of Urban Garbage Levels Using Time-of-Flight Sensing Technology," *Sensors*, vol. 25, no. 7, 2025, doi: 10.3390/s25072152.
- [16] W. E. F. Anggara, H. Yuana, and W. D. Puspitasari, "Rancang Bangun Alat Monitor Ketinggian Air Berbasis Internet of Things (IoT) Menggunakan ESP32 dan Framework Blynk," *JATI (Jurnal Mhs. Tek. Inform.)*, vol. 7, no. 5, pp. 3837–3845, 2024, doi: 10.36040/jati.v7i5.7956.
- [17] Y. A. Muhammad Yazid and R. A. Permana, "Rancang Bangun Prototype Monitoring Lampu Jalan Secara Otomatis Menggunakan Mikrokontroler ESP32 dan API Bot Telegram," *J. Tek. Inform.*, vol. 8, no. 1, pp. 12–19, 2022, doi: 10.51998/jti.v8i1.477.
- [18] Y. Ratmini, V. Atina, and E. Purwanto, "Sistem Monitoring dan Peringatan Dini Banjir Berbasis Internet of Things (IoT)," *J. Ilm. Teknol. Inf. Asia*, vol. 19, pp. 1–8, 2025.
- [19] D. Azizi and V. Arinal, "Sistem Monitoring Daya Listrik Menggunakan Internet of Thing (IoT) Berbasis Mobile," *J. Indones. Manaj. Inform. dan Komun.*, vol. 4, no. 3, pp. 1808–1813, 2023, doi: 10.35870/jimik.v4i3.409.