eISSN: 2477-7994

JMECS (Journal of Measurements, Electronics, Communications, and Systems)
Vol.11, No.1, Pg.1-10, June 2024

IoT-Based System for Pothole Mapping

Muhammad Andika Naufan Dwiandharu[1*], Danny Aulia[1], William Wafi Wisnutama[1], Aloysius Adya Pramudita[2], Harfan Hian Ryanu[2], Vinsensius Sigit Widhi Prabowo[2]

[1]*School of Electrical Engineering, Telkom University, Bandung, 40287, Indonesia*
[2]*The University Center of Excellence Intelligent Sensing-IoT, Telkom University, Indonesia.*
[*]andikanaufan@student.telkomuniversity.ac.id
Manuscript received March 30, 2024; revised May 16, 2024; accepted June 7, 2024

Abstract

Advancements in technology have enabled various innovations that increase convenience and improve services. One such innovation is the Internet of Things (IoT). In this research, we developed a system to detect and map road damage. The goal is to provide users with information about the condition of nearby roads. This project is part of developing IoT-based location mapping systems and highlights the importance of continuing advancements in mapping and detection technologies. The detection process uses an accelerometer sensor to measure changes in an object's axis caused by vibrations. During pothole detection, the accelerometer records acceleration values along the x, y, and z axes by tracking changes in axis values for each event. The data is captured by an Arduino Nano and sent to a Raspberry Pi for processing, which also retrieves data from a GPS and camera before transmitting it. When the data meets the classification criteria, the GPS collects coordinate data, and the camera captures images. This processed information is then sent to a database and displayed on an application dashboard. The system was tested along Jalan Radio Palasari in Bandung Regency, where 50 data points were collected with the sensor's detection delay to 1 minute, while data transmission to the database took 6 seconds. The detection and mapping accuracy was approximately 85%, with a location variance of only 3-6 meters from the actual pothole. The results of this study can help drivers avoid damaged roads, and future improvements in camera quality and internet connectivity could further enhance accuracy.

Keywords: *Damaged Road, internet of Things, Mapping Systems, Sensor Detection*

DOI : 10.25124/jmecs.v11i1.7666

1. Introduction

Over time, roads will experience a decline in quality due to the daily load from passing vehicles. Careful planning in road construction is essential to ensure that maintenance can be performed effectively. While the load factor may be the main factor of concern, several other factors must be considered, such as changing weather conditions or the drainage system owned by the road. Indonesia, as a country with a relatively extreme climate, needs to pay attention to road maintenance [1 - 3]. We present a comprehensive review of the updates to our study, introducing a novel approach to road damage detection and visualization by integrating accelerometer data with GPS and cameras. This method enhances the accuracy and mapping of road damage compared to previous studies. With this contribution, we aim to demonstrate the innovative aspects of our research.

Road damage is a common issue in Indonesia, with 57.97% of roads affected [4]. Damaged roads can lead to increased traffic congestion and longer travel times. Additionally, they disrupt the delivery of goods, resulting in suboptimal performance and delays. [5].

Comparing the quality of highways using the Road Quality Index (WRI), Indonesia scores 4.20, placing it below other ASEAN countries like Malaysia and Singapore. Singapore, with a QRI score of 6.50, holds the top position, followed by the Netherlands. Countries with high QRI values tend to have higher transportation speeds. Countries such as Canada, Saudi Arabia, and Portugal have the highest vehicle speeds. Having a high QRI value makes better use of road services to maximize output.

Enhancing data accuracy in road damage detection and visualization can help address this problem. Previous research [6], [7] utilized Deep Learning with stationary cameras. This research has limitations in the data variation section since it only showed data where the camera was placed. Another method used was the R-CNN (Regional Convolutional Neural Network) [8], [9] method. This method represents an evolution of the Convolutional Neural Network (CNN) approach. However, the R-CNN results are still not optimal, as several points were incorrectly detected, which decreased the overall accuracy.

Our study aims to detect road damage and data accuracy and data visualization using an accelerometer and GPS. This study incorporated specific algorithms into the system. The added algorithm generates a threshold value, which is used then to classify different types of shocks by setting value limits for each data classification. The processed information data will display road damage information and detected location points. For this purpose, we installed sensors in public transportation vehicles that continuously detected road damage throughout their operation until they stopped. The type of data we detect is classified into three types: good road condition, intermediate potholes, and severe potholes. All detection system results are stored in a cloud database to aid in data visualization. The Internet of Things (IoT) was utilized to detect road damage and transmit information quickly. Later, data collection was carried out using several public vehicles which reported the condition of the road every day. Unlike private vehicles which have the privilege to avoid damaging road, public transportation should follow the determined route. Hence, placing sensors in public transport reduced the amount of road damage data that went unreported.

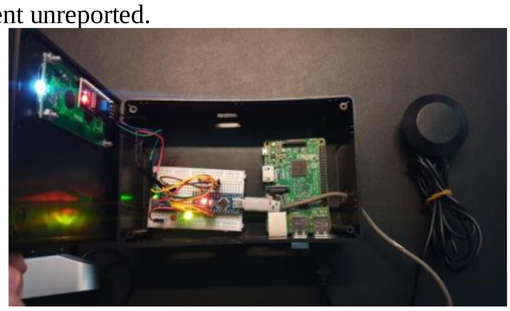

Fig. 1. Inside View of The Device

2. Research Method

A. System Design

The solution system is a hardware and software system using an accelerometer, global positioning system (GPS), and camera sensors. Fig. 1 shows an accelerometer mounted with an Arduino Nano on a breadboard. The sensor requires adequate protection, so it is housed in an additional case measuring 18.3 cm in length, 11.3 cm width, and 6 cm heigh. Fig. 2 shows the wiring diagram of the hardware used. The wiring diagram shows the connection between the Raspberry Pi and the sensor you are using.

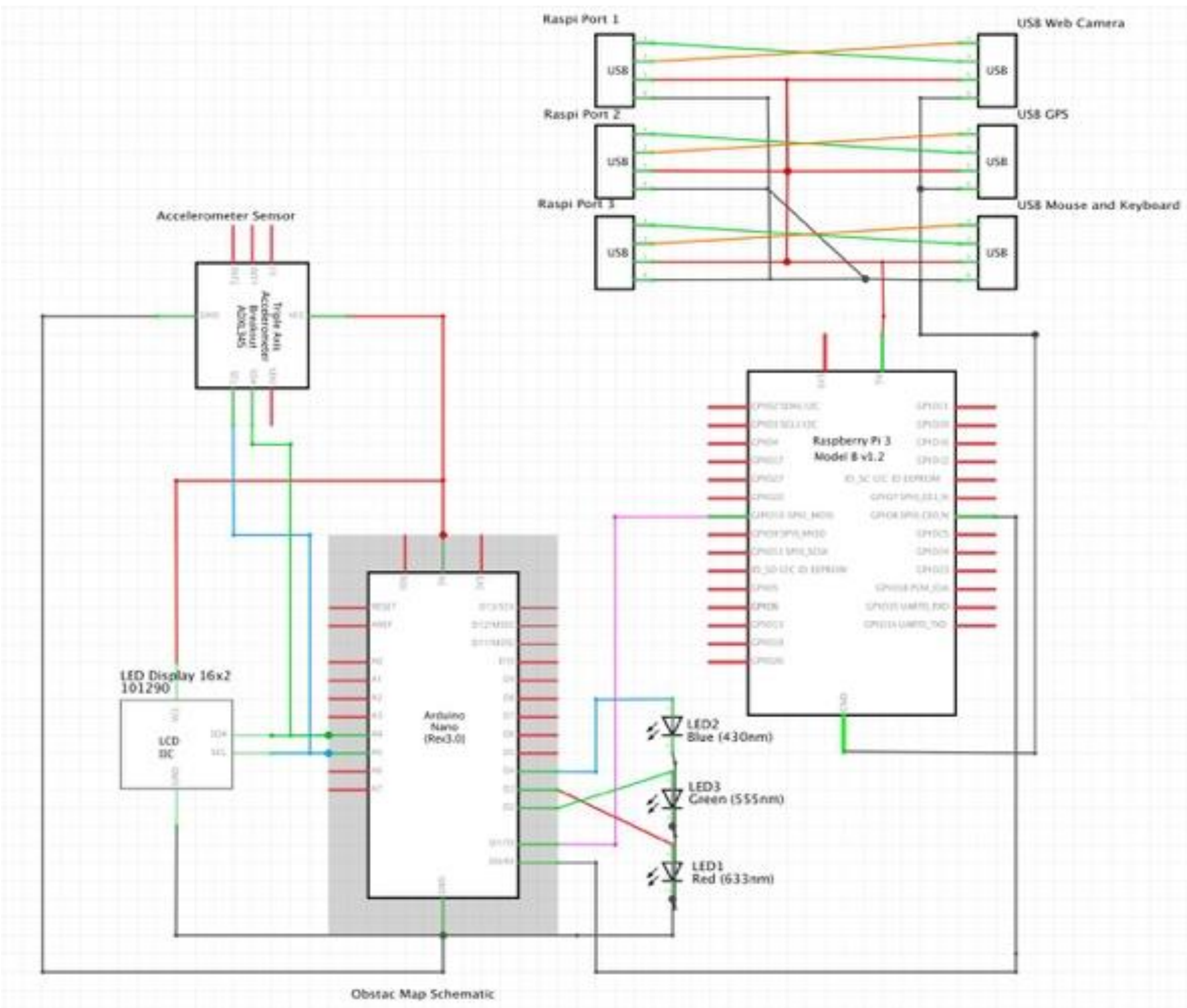

Fig. 2. Wiring Diagram System

The Arduino microcontroller is an open-source board designed to interact with sensors in the surrounding environment [10]. It is programmed using C or C++. To update or configure the Arduino, it can be connected to a laptop via a USB cable. The programming process is done through the Arduino IDE, a user-friendly platform compatible with most computers.

The Arduino device we use is Arduino Nano. The microcontroller used in Arduino Nano shown in Fig. 5 is ATMega328P. Arduino Nano has a large area of 18 x 45 mm and weighs 7 grams. It can operate on 5 V power. Arduino nano has 22 ports, 8 of which act as ports for analog input and 14 of which are used as digital I/O ports [11],[12]. Arduino Nano has a CPU speed of 16 MHz, enabling it to perform 16 million operations per second. To program the Arduino Nano, the Arduino IDE is used, selecting the appropriate port and the ATMega328P microcontroller.

Using references from [13], the research utilized ActiGraph GT3X+ accelerometers. The model already has a cover that allows each end to be connected like a clock or belt. The accelerometer measures 19 x 24 x 45 mm and weighs 19 grams. It is also resistant to water pressure to a depth of 1 meter [14]. The accelerometer can detect motion in three directions along the x, y, and z axes. It operates at a frequency of 30 Hz, allowing it to capture data 30 times every second. The accelerometer can also measure acceleration data from -2g to +2g (1g = 9.81) [15].

As you can see in Fig. 3 the ADXL345 is a microcontroller designed to perform 3-dimensional speed calculations (using 3 axes). It is small and thin, measuring 3mm x 5mm x 1mm, making it ideal for portable devices. This sensor is also low-power, making it suitable for battery-operated devices or those used for extended periods.

The sensor's power consumption must be taken into account. If the power usage is high, a correspondingly powerful energy source will be required. If the source power does not match the sensor requirements, it could damage the sensor, the power source, and even the connecting cable. The damage that appears delays retrieving data because the type of damage must be identified first before repairing the damaged part.

In this research, the ADXL345 sensor is used to detect vibrations when passing over the road. Before collecting data, the sensor requires time to calibrate the system. Calibration ensures that the sensor output does not rely on previous values. This process neutralizes the sensor's readings and displays the sensor's offset value. If the sensor uses previous data, the output from the accelerometer may not accurately reflect the road conditions, leading to errors in readings. As explained in the block diagram of this sensor output, there are three types of thresholds applied in each part.

This sensor will be placed on the breadboard along with the Arduino Nano to ensure it remains stable. By placing the sensor in the same box as the Arduino, the accelerometer detects the shock experience by the box.

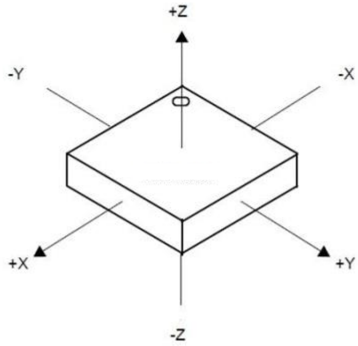

Fig. 3. Accelerometer Axis

additionally, housing the accelerometer in the box offers better protection compared to leaving it exposed. Data collection was conducted using a Beat motorcycle type D1B02N12L2, in 2018. During data collection was conducted the motorcycle was driven at a speed of approximately 25-30 km/h. The equipment was placed on the passenger bench where detection is carried out.

Research on animal theft tracking systems using Arduino and GPS modules [16] highlights the importance of GPS technology. GPS is a system that uses sensors to receive signals from multiple GPS satellites orbiting the Earth at all times. At least four satellites are required to determine the user's GPS location. The signals received from these satellites are processed using triangulation techniques to provide the user's GPS coordinates.

An example of this type of GPS is the NEO-7M, which is designed to detect L1C/A (Coarse/Acquisition) signals at a frequency of 1575.42 Mhz. The L1C/A [17], [18] is available for public use and is a standard signal commonly used. While this signal provides fairly good accuracy, as the name suggests, 'coarse,' it's GPS readings are less precise compared to military signals like P(Y), where the 'P' stands for precise.

Our group uses the Ublox Neo-6M GPS, which is designed to connect to a computer via a USB cable for data reading. The GPS takes about 25 - 30 seconds in a stationary position before producing an output. This delay occurs because the GPS searches for multiple satellite sources to provide data. Once it gathers sufficient data, it performs triangulation to calculate and output latitude and longitude coordinates. If no data is provided or no output is generated, it's likely that the GPS is indoors, preventing it from detecting satellites and performing triangulation. GPS damage can also cause detection failure. In some types of GPS, damage may be indicated by the absence of indicator lights [19].

The information regarding latitude and longitude is provided in the GPGGA (Global Positioning System Fix Data) code, with the coordinates still in DDM (Degrees Minutes) format. The latitude data must be converted into DD (Decimal Degree) for use in the application, making it easier to understand. The steps to convert the data can be seen below in Eq (1),

$$DD = Degree + \left(\frac{Minutes}{60}\right) \quad (1)$$

The GPS will be positioned outside the box, specifically on top of the lid. Its external placement allows the GPS to receive satellite signals, which is crucial for providing accurate data output. If the GPS is placed inside the box, there is a risk of interference that could reduce its accuracy.

To detect significant road damage, an image of the identified damage will be captured. A web camera will be used for this purpose, positioned at the rear of the vehicle and directed downwards. Additionally, the camera's data collection will include a delay of about 3 seconds to allow the camera time to gather adequate light.

Raspberry Pi serves as the platform for system integration. Also known simply as Raspberry, it is a compact version of modern computers, using different types of processors that allow for the installation of open-source operating systems. Raspberry Pi supports various types of programming, such as Python, C, C++, BASIC, and Ruby [20].

Raspberry Pi 3b+ features a Broadcom BCM2837B0 CPU with a quad-core speed of 1.4 GHz and 1 GB of RAM. it offers three options for network connectivity: wireless fidelity, Bluetooth, or an ethernet cable [21]. The device uses a micro SD card with a capacity of 16 GB, which comes preloaded with the operating system compatible with the Raspberry Pi.

In this research, Raspberry Pi is used as a place for sensor integration, working alongside Arduino, GPS, and a camera. Sensor integration is performed using the Thonny IDE in a Python environment. The condition given is that when the Arduino detects damage the Arduino retrieves the converted latitude and longitude data from the GPS and take pictures for the primary damage condition. The data is stored in JSON formatted files. The output includes three types of JSON files, each named according to the type of damage detected. Each JSON file contains three types of data sent to Firebase: latitude, longitude (as text), and an image of the damage. The image is stored in base64 format so that storage is not too heavy. The base64 format is only temporary because at the time of sending the base64 file type is converted to its original type. JSON files were chosen to make it easier to send to Firebase.

The data retrieval process is done using a python script by using a virtual environment to retrieve the required dependencies. The data retrieval process will use a looping process where data will be sent every 10 minutes and after sending back to retrieve data. This process will continue to be repeated until the script is stopped directly by the user.

B. Block Diagram

Fig. 4, shows our research diagram block consisting of the hardware and software parts. The hardware part shows how each sensor works on the system used, such as the accelerometer and GPS, and the integration of the sensors using a USB cable to the Raspberry Pi. In the software section, data is collected from each sensor using the programming language on the Raspberry Pi and sent to Firebase. The software also provides data visualization using a mobile application.

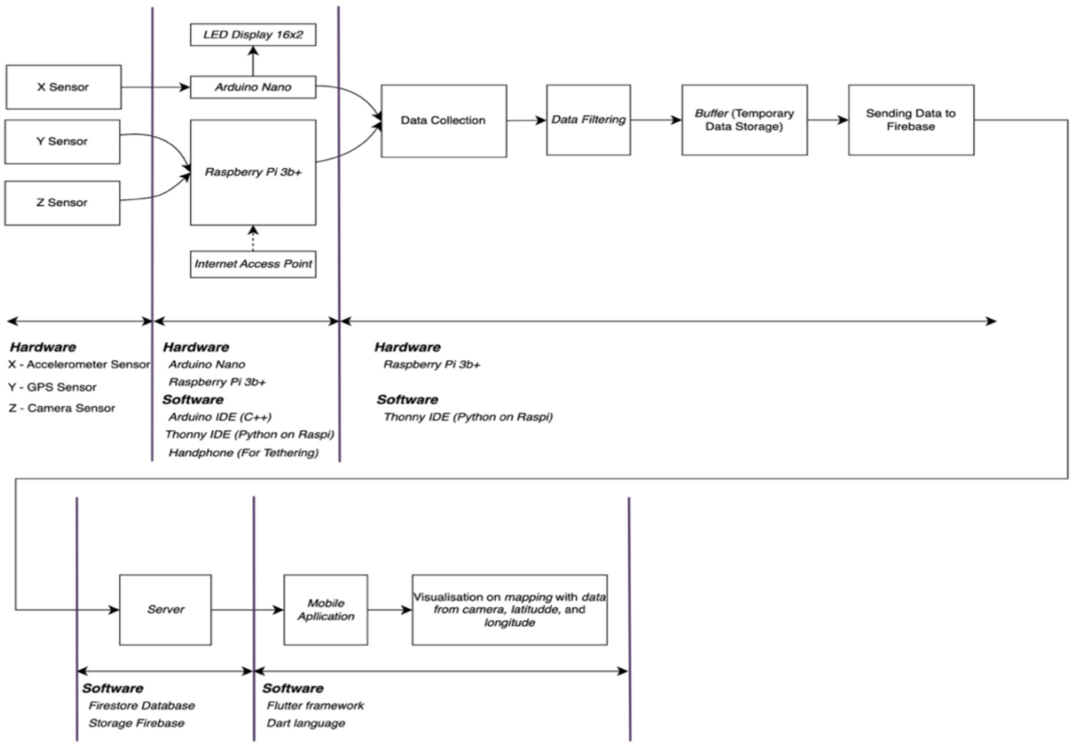

Fig. 4. Block Diagram

Table 1. Road Damage Classification Table

Depth of Road Potholes	Length of Road Potholes		
	102 – 203 mm	203 – 457 mm	457 – 762 mm
12.7 - 25.4 mm	L	L	M
25.4 - 50.8 mm	L	M	H
> 50.8 mm	M	M	H

L : No need to patch road; M : Partial patching on potholes; H : Complete patching on potholes

Accelerometers are sensors that can detect vibrations on three axes (x,y,and z) The accelerometer used a limit of 2g which will give an output ranging from -32.810 to 32.810. A classification table from a previous study [22] that researched road damage to determine the damage classification. Table 1 shows the classification of road damagebased on the depth and length.

Based on the table, you divide the road condition classification into three classes, namely roads that do not need repair, partial patching, and patching at all depths. The classification is supported by other aspects such as the depth of the pothole and the diameter of the road damage. The three classifications of road conditions also follow the standards made by the U.S. Army Corp of Engineers. Where they have assessed the condition of pavement damage called the Pavement Condition Index (PCI) for the needs of airport pavements, roads, and parking lots has been widely used in America. Based on this classification, we created three road condition classifications following the existing parameters. Good road condition follows the parameters of a road with a depth of 12.7 - 25.4 mm, minor deterioration with a depth of 25.4 - 50.8 mm, and major deterioration with a depth of more than 50.8 mm. The results of vibration detection by the accelerometer are classified as follows:

- Good condition roads ($0 \leq Vibration\ value \leq 3000$)
- Intermediate potholes are shown in Fig. 5 ($3000 < Vibration\ Value \leq 6000$) with pothole depths ranging from three centimeters to five centimeters.
- Severe potholes are shown in Fig. 6 ($Vibration\ Value > 6000$) with pothole sizes starting from five centimeters onwards.

In the classification of damage types, two types of thresholds are used, with the first threshold worth 3000 and the second threshold worth 5000. The code is first inserted into the Arduino using the Arduino IDE application. The reading results from the Arduino provides the data latitude and longitude that have been taken from the GPS. To save the file to storage, the output data is saved to the respective files according to the type of damage detected. Especially for significant damage, images are also taken to add data visualization. Before the storage is saved permanently, the documents number is checked first to avoid any duplicates. Data with the same numbering can cause a lot of trouble, mainly overwriting the data before and losing it in the process.

The data that have been collected by using the sensor are going to be sent to Firebase to the Firestore section for text type data and to the storage section for jpeg data. To be able to send the data, Raspberry Pi needs an internet connection to send data; this problem can be solved by using a Wi-Fi module. The Wi-Fi module used by Raspberries functions so that Raspberries can send data to Firebase. The data that has been sent to Firebase is then used in visualization using a mobile application.

C. System Testing Method

Testing on the accelerometer has been conducted twice. The first test aims to retrieve unfiltered data from the accelerometer. This raw data is used to determine the threshold value for classifying the type of road damage encountered. Before data collection, the types of damage to be observed were established. Light damage is defined as having a hole with a depth of at least 3 cm, while heavy damage is defined as having a hole depth of at least 5 cm, and so on. Determining the classification of hole depth is important. Because the depth of the hole will have a lot of influence on the vibrations received by the accelerometer

From the test we conducted, it is evident that when passing through moderate potholes, the accelerometer data for mild damage shows the x-axis values ranging from 2000 to over 4000. In contrast, for severe potholes, the y-axis values indicate detection levels above 5000. Based on the results of the data collection, it was decided that two types of thresholds would be used. The first threshold is at 2000, and the second is at 5000. Using these thresholds, the accelerometer can indicate moderate potholes when the readings are between 2000 and 3000, and severe potholes when the readings exceed 5000. The accelerometer is set to give output data every second; when the numbers are

Fig. 5. Intermediate Pothole

Fig. 6. Severe Pothole

average, it provides an output saying that the road is in good condition.

The results of the accelerometer readings will be stored in JSON, with each type of damage given a distinct name based on the detected condition. For each new data entry stored in JSON, an incremented number is added to the dataset.

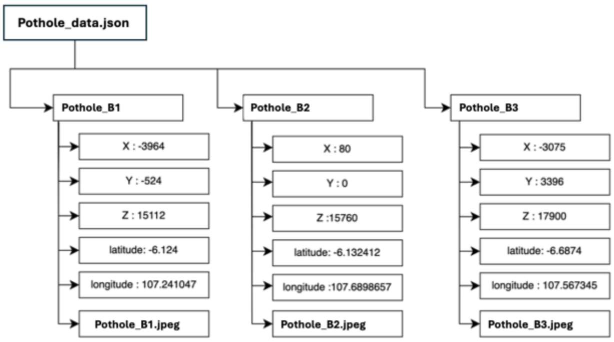

Fig. 7. JSON Structure

Fig. 7 shows you the JSON file with Indonesian names to match the output of the sensors we use, for severe potholes we give the name kerusakan_besar_data.json, intermediate potholes are given the namekerusakan_kecil_data.json, and for good road condition we give the name kondisi_null_data.json. Each data detected in each damage will be given a code name in Indonesian with the naming "Kerusakan_(Kode)(Nomor)". The code for severe potholes uses the code "B", for intermediate potholes uses the code "R", and for good road conditions uses the code "N". The number will have an increment every time new data is entered.

Sending the data using JSON is done using the Python programming language on a Raspberry Pi device. Using Python modules, we can send the data by adding a JSON file that stores the Firebase API to connect to the Firebase itself. The JSON file can be downloaded from the Firebase setting. After adding the file, you add a link that resides in Firebase and goes to your Firebase to store the text in the Firestore database; to send the image, you can use a module named bucket to send an image to your Firebase storage. Data obtained through the accelerometer is forwarded to the Raspberry Pi so that it can be sent to the database using the Wi-Fi module. In testing the data transmission to the database, data is sent five times, with readings taken and sent every 60 seconds. This 60-second interval has been calculated to account for all conditions, including a 3-second delay for significant damage within the total capture time. At the end of each 60 seconds, all types of data are sent to Firebase, namely major, minor, and no damage re sent to Firebase.

3. Result and Discussion

A. Accelerometer Sensor

We tested the accelerometer sensor on the road behind Telkom University. The purpose of this test was to verify that the threshold value and logic gate used are appropriate and can produce the desired road classification output. As explained in the previous discussion, we established 3 types of road classifications based on standards set by the U.S. Army Corps of Engineers. To determine the three types of road classification, two influential aspects are considered: the depth of the pothole and the length of the pothole traveled. The first stage of the classification is to read the z value to determine the class of depth data read by the accelerometer. Then, the next classification calculates the delta S value from the x and y deltas detected by the accelerometer. The delta s value obtained is converted into mm units, this is to facilitate the making of logic gates because the table in the previous discussion uses mm units in determining the condition of the hole length. After that, the accelerometer will provide output in the form of the type of road classification obtained and also the number of each axis obtained. The following is a table of results from accelerometer testing.

Table 2. Different Data Type from Data Collection

Z Axis (i)	Z Axis (i+1)	ΔZ	S (Distance Traveled)
Good Road Classification			
17436	17596	160	100.0 mm
17396	17592	-280	224.0 mm
17592	17716	124	300.0 mm
Intermediate Potholes Classification			
24120	14448	9672	439.3 mm
16480	22900	6420	392.9 mm
14304	21472	7168	186.002 mm
Severe Potholes Classification			
5924	17492	11568	931.9 mm
18416	9868	8548	750 mm
16948	30404	13456	654.5 mm

The data in the table above shows the classification of road conditions based on the shocks received by the accelerometer sensor. The test was conducted using a motorcycle with an average speed of 25 km/h. During the test, we got 336 data detected by the accelerometer. We took 15 random data from each predefined road classification. The negative values on the accelerometer axis indicate the direction and magnitude of the shocks detected by the sensor. After we analysed the 45 road damage sample data, we found that 3 roads did not match the classification. Based on this, we tried to determine the percentage of threshold success by calculating the error data obtained from a total of 45 samples taken. From the sample data taken, 42 road condition classification data are under the existing grouping. So the percentage obtained from accelerometer testing reaches 93.33% of the total test data.

B. GPS Sensor

Testing of the GPS and accelerometer sensors was conducted on Jalan Radio, directly behind Telkom University. The accuracy of the GPS is evaluated by visualizing the reading results with an application. The visualization results from the application can show where the GPS is read, and where each point is represented by the data results issued by the GPS. The actual position can be compared with the position shown by the GPS based on the traveled road. In this project's data collection, the left lane was used as an example. Table 2 shows the difference between the initial GPS data in the form of DDM and the data after conversion in the form of DD.

Table 3 Data Discrepancy Between GPS and Original Position

No.	Latitude & Longitude of GPS (DDM Format)	Latitude & Longitude of GPS (DD Format)	Distance Difference
1	-6° 58.3781' , 107° 37.5962'	-6.972980166666668, 107.62658583333332	5 m
2	-6° 58.3789' , 107° 37.5829'	-6.972972, 107.62638783333333	5 m
3	-6° 58.3734' , 107° 37.5324'	-6.972935166666668, 107.62553766666666	5 m
4	-6° 58.3751' , 107° 37.5182'	-6.972924166666667, 107.6252595	6 m
5	-6° 58.3717' , 107° 37.4881'	-6.9728995, 107.62479183333335	5 m
6	-6° 58.3726' , 107° 37.4833'	-6.9728935, 107.62471816666667	3 m
7	-6° 58.3708' , 107° 37.4670'	-6.972883333333334, 107.62440399999998	6 m

Observations on Jalan Radio behind Telkom University resulted in 171 data points generated by the GPS. We calculated the difference in distance between the GPS readings and the actual conditions of the road damage. The calculation process is done by selecting 10 random samples from the difference in GPS points detected. It was determined that the average discrepancy between the GPS-measured distance and the actual road damage condition was 5.7 meters. we also found that the average difference between the GPS distance obtained and the actual condition of road damage reached around 3-6 meters. This discrepancy may be due to the type of GPS data captured, as the data generated is not as accurate as the data detected by military-grade systems. Improving GPS accuracy can be done by adding receivers and antennas that have better quality than those of GPS. This includes more sensitive antennas and receivers, as well as enhanced selectivity and multipath rejection capabilities. Another alternative is the multi-GNSS integration method that combines data from Global Navigation Satellite Systems such as GLONASS, Galileo, and BeiDou with GPS. Adding more satellites can improve GPS accuracy and reliability, especially in extreme conditions.

C. Sending Data to Database

The second test is conducted after the accelerometer data has been successfully filtered from the first test results. This second test can be performed simultaneously with the GPS test. The detection results show 171 spots, with 7 spots as severe potholes, 41 spots as moderate potholes, and 123 spots indicating the road is passable. The second test is conducted after the accelerometer data has been successfully filtered from the first test results. This second test can be performed simultaneously with the GPS test.

The data collection experiment described in the previous section will produce two types of data. The first set of data concerns the time at which the data is collected, with the time being recorded by the observer. The second data will show the length of data transmission. The time calculated in the second data collection is the time when the system stops taking data until the system recalibrates. Table 4 shows the time taken during data collection for each loop.

Table 4 Time Taken During Data Retrieval Experiment

No	Length of Time Taken
1	1:05.22
2	0:59.72
3	0:58.97
4	0:58.65
5	0:59.02

From the results of the data retrieval and data transmission experiments, the data that should have been taken every minute when recorded was not exactly one minute in the retrieval process. From the results of collecting data five times, four readings were under 1 minute and one data was above 1 minute, while one was above 1 minute. This is not in accordance with the ideal conditions specified in the script used. The difference in data retrieval time can occur due to observer error in retrieving data. Every time data is sent, the time taken to send it is calculated. To see more efficiency, you can see the Table 5.

The data collection experiment described in the previous section will produce two types of data. The first set of data concerns the time at which the data is collected, with the time being recorded by the observer. This can happen because the JSON file used as the first storage continuously accumulates the retrieved data when the system retrieves the data. To optimize the image type file, we first convert it into text and zip the image so that it does not burden the storage owned by the device. Zipping the file will change the file size and make it easier to send to databases.

Table 5 Trial on Sending Data to Firebase

No	Start Sending	Finish Sending	Transmission Time
1	1:05.22	1:08.49	0:03.27
2	2:08.21	2:13.81	0:05.60
3	3:12.78	3:19.32	0:06.54
4	4:17.97	4:26.46	0:08.49
5	5:25.48	5:35.32	0:09.84

Another factor that can cause delays is the network performance at the time. Slow network speeds can prolong data transmission. The current upload speed is 13.7 Mbps, which is relatively high. Reducing interference or using an optimal frequency can further increase upload speed. A faster upload speed can help to send data faster so that less data is lost when the code sends data to Firebase.

In addition to internet capabilities, human error can also be a factor in the error of data collection. Data collection is carried out only using a stopwatch that must be operated by humans, making the margin of error in data collection even greater.

4. Conclusions

This research offers data visualization through a mobile application, based on the hardware system installed on the vehicle. Based on 10 random GPS locations, we found the average distance difference between the actual hole position with GPS coordinates reached 5.7 meters. In addition, the accelerometer sensor has an accuracy of 93.33% which has been tested from 45 random sample data. The accuracy can be increased by optimizing the sensor and the classification method used in the data collection process.

References

[1] A. A. Siregar and L. Cassiophea, "THE EFFECT OF VEHICLE VOLUME ON DAMAGE TO PINE ROAD IN PALANGKA RAYA CITY," 2023. Accessed: Jun. 05, 2024. [Online]. Available: https://doi.org/10.37304/parentas.v8i1.4364

[2] J. T. Berkelanjutan and A. G. Feriko, "ANALISIS KERUSAKAN JALAN DI RUAS SIMPANG BERENG BENGKEL TUMBANG NUSA," *Sustainable Technology Journal*, vol. 8, no. 2, pp. 77–81, 2019, [Online]. Available: http://jtb.ulm.ac.id/index.php/JTB

[3] W. B. Mestuni and N. Hartatik, "ROAD DAMAGE EVALUATION ON RIGID PAVEMENT WITH PAVEMENT CONDITION INDEX METHOD IN KLAKAH REJO-BENOWO ROAD, SURABAYA CITY, EAST JAVA," *Jurnal Teknik Sipil*, vol. 23, no. 4, p. 635, Feb. 2024, doi: 10.26418/jts.v23i4.61048.

[4] Efliza, W. Suska, Habibulloh, A. Zahidi, and H. Wulandari, "Land Transportation Statistics," 2022, Accessed: Jun. 15, 2024. [Online]. Available: https://www.bps.go.id/id/publication/2023/11/27/5a5e4c75e4a25d44b1846446/statistik-transportasi-darat-2022.html

[5] J. Wen, Z. He, Y. Yang, and Y. Cheng, "Study on the factors and management strategy of traffic block incident on hangzhou province highway," in *Proceedings - 2020 International Conference on Intelligent Transportation, Big Data and Smart City, ICITBS 2020*, Institute of Electrical and Electronics Engineers Inc., Jan. 2020, pp. 67–71. doi: 10.1109/ICITBS49701.2020.00022.

[6] S. D. Cahyono, "SISTEM DETEKSI KERUSAKAN PERMUKAAN JALAN RAYA DENGAN METODE TEMPLATE MATCHING," *Penelitian Ilmu-Ilmu Ekstata*, vol. 15, pp. 105–111, 2014, Accessed: Aug. 06, 2024. [Online]. Available: https://unmermadiun.ac.id/ejurnal/index.php/agritek/article/view/108/202

[7] J. Zhao, L. Zhang, J. Ye, and C. Xu, "MDLF: A Multi-View-Based Deep Learning Framework for Individual Trip Destination Prediction in Public Transportation Systems," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 8, pp. 13316–13329, Aug. 2022, doi: 10.1109/TITS.2021.3123342.

[8] A. M. Qureshi, N. Al Mudawi, M. Alonazi, S. A. Chelloug, and J. Park, "Road Traffic Monitoring from Aerial Images Using Template Matching and Invariant Features," *Computers, Materials and Continua*, vol. 78, no. 3, pp. 3683–3701, 2024, doi: 10.32604/cmc.2024.043611.

[9] M. B. Asad, F. J. Sharna, N. Alam, and M. J. Islam, "HSV and Template Matching Based Bengali Road Sign Recognition Technique," 2016. Accessed: Jun. 05, 2024. [Online]. Available: 10.1109/ICISET.2016.7856527

[10] D. Ajiatmo, F. A. Malik, and A. Raikhani, "Desain Lampu Cerdas Berbasis Arduino Nano Untuk Kondisi Bencana," *BEES: Bulletin of Electrical and Electronics Engineering*, vol. 4, no. 2, pp. 42–49, Nov. 2023, doi: 10.47065/bees.v4i2.4302.

[11] A. Shakirovich Ismailov Zafar Botirovich Jo, "Study of arduino microcontroller board," Mar. 2022. Accessed: Jun. 05, 2024. [Online]. Available: https://www.researchgate.net/publication/359502443_Study_of_arduino_microcontroller_board

[12] M. Hasyim and A. Ahfas, "Revolutionizing Distance Measurement with Arduino Nano," *Indonesian Journal of Innovation Studies*, vol. 25, no. 2, Apr. 2024, doi: 10.21070/ijins.v25i2.1138.

[13] J. Skotte, M. Korshøj, J. Kristiansen, C. Hanisch, and A. Holtermann, "Detection of physical activity types using triaxial accelerometers," *J Phys Act Health*, vol. 11, no. 1, pp. 76–84, 2014, doi: 10.1123/jpah.2011-0347.

[14] R. and S. D. ActiGraph, "User Guide ActiGraph GT9X Link + ActiLife," 2020. Accessed: Jun. 05, 2024. [Online]. Available: https://s3.amazonaws.com/actigraphcorp.com/wp-content/uploads/2020/03/05155628/ActiGraph_Link_UserGuide_E.200.6001_Revision6_FINAL.pdf

[15] A. Aydemir and A. Tafyun, *SELF-PACKAGED THREE AXIS CAPACITIVE MEMS ACCELEROMETER*. IEEE, 2020. Accessed: Jun. 05, 2024. [Online]. Available: 10.1109/MEMS46641.2020.9056192

[16] G. Ramesh, K. Sivaraman, V. Subramani, P. Y. Vignesh, and S. V. V. Bhogachari, "Farm Animal Location Tracking System Using Arduino and GPS Module," in *2021 International Conference on Computer Communication and Informatics, ICCCI 2021*, Institute of Electrical and Electronics Engineers Inc., Jan. 2021. doi: 10.1109/ICCCI50826.2021.9402610.

[17] Z. Jiao, Y. Yang, H. Zhu, and F. Ren, *Realization and Improvement of Object Recognition System on Raspberry Pi 3B+*. IEEE, 2018. Accessed: Jun. 05,

2024. [Online]. Available: 10.1109/NAFOSTED.2017.8108066

[18] Universitas Gadjah Mada, Institute of Electrical and Electronics Engineers. Indonesia Section., and Institute of Electrical and Electronics Engineers, *Evaluation on the performance of NTRIP RTK positioning using multi-frequency low-cost GPS module in areas with different telecommunication signal strength*. 2019. Accessed: Jun. 05, 2024. [Online]. Available: 10.1109/TITS.2021.3123342

[19] L. Deng, H. Guo, N. Xiros, and M. Yu, *A Research on Roll Angle Calculations Based on IMU/GPS Compass for Ships*. 2023. Accessed: Jun. 05, 2024. [Online]. Available: 10.1109/PLANS.2016.7479797

[20] H. Dipak Ghael, L. Solanki, G. Sahu, and A. Professor, "A Review Paper on Raspberry Pi and its Applications," *International Journal of Advances in Engineering and Management (IJAEM*, vol. 2, p. 225, 2008, doi: 10.35629/5252-0212225227.

[21] T. Le-Tien, T. Huynh-Kha, T. Nguyen-Thanh, T. Pham-Duc, and T. Nguyen-Huu, *An Evaluation on the Effective Processing Time of Raspberry Pi-3B-based Embedded Systems for Determining Forged Images*. 2017. Accessed: Jun. 05, 2024. [Online]. Available: 10.1109/NAFOSTED.2017.8108066

[22] W. Widodo, A. Rahmawati, and E. Adly, "MODEL ANTISIPATIF MENGATASI KERUSAKAN PERKERASAN JALAN DI KABUPATEN BANTUL YOGYAKARTA," 2017. Accessed: Jul. 04, 2024. [Online]. Available: https://repository.umy.ac.id/handle/123456789/14066

Author Information

Muhammad Andika Naufan Dwiandharu was born in 2002 in Bandung, Indonesia. He is currently completing his undergraduate degree in Telecommunication Engineering. Previously, Andika was also the head of assistant laboratory of Basic Physics I and II, as well as the coordinator of Cryptography and Security Protocols practicum.

Danny Aulia was born in 2001 in Jakarta, Indonesia. He is currently studying at Telkom University in the program of Telecommunication Engineering. As a student, Danny is also active as an assistant for the research of Signal Processing for Intelligent System and an assistant for the practicum of Cryptography and Security Protocol.

William Wafi Wisnutama, born on May 15, 2003, in Bandung, Indonesia, is currently pursuing a Bachelor's degree in Telecommunications Engineering at Telkom University. In addition to his studies there, he is also enrolled at Universitas Pendidikan Indonesia. William actively contributes as a laboratory assistant in both the Basic Computer Laboratory and the Cryptography & Security Protocols Laboratory. His dual roles in academia and practical lab work reflect his commitment to the field of telecommunications and his dedication to advancing his technical skills and knowledge.

Prof. Aloysius Adya Pramudita, a pioneer in the world of Telecommunication Engineering, brings more than two decades of academic experience. Holding the title of professor, he now leads as Director of the Center for Excellence in Innovation and Technology Development (PUI-PT). His outstanding expertise in Radar, Electromagnetic Applications, and Propagation Antennas has inspired a new generation in these fields. As an accomplished teacher and researcher, Prof. Adya Pramudita continues to guide and encourage innovation in the world of telecommunication technology.

Harfan Hian Ryanu received the B.S. degree in telecommunication engineering from the Telkom Institute of Technology, Bandung, Indonesia, in 2011, and the M.E. degree in telecommunication engineering from Sydney University, Sydney, NSW, Australia, in 2018.,He is currently a Lecturer with the School of Electrical Engineering, Telkom University, Bandung. He is also with the Basic Transmission Laboratory. He is interested in RF devices, especially in the remote sensing areas, radar, and antenna; and has published some articles in those fields in national journals.

Vinsesius Sigit Widhi Prabowo, S.T., M.T., is an expert assistant and researcher who is actively involved in PUI-PT, with a special focus on the application of differential equations in various applications. Starting his higher education at Telekom University, Vincent successfully obtained his Bachelor of Engineering (S.T.) degree in 2015 and continued to earn his Master of Engineering (M.T.) degree at the same institution in 2017.

Additional Information