

An IoT-Based Decision Support System for Real-Time Vehicle Maintenance Scheduling in the Fourth Industrial Revolution

1st Diky Siswanto

Dept. of Electrical Engineering
Universitas Widyagama Malang
Malang, Indonesia

dsiswanto@widyagama.ac.id;

<https://orcid.org/0009-0002-2491-9065>

2nd Akhmad Farid

Dept. of Mechanical Eng.
Universitas Widyagama Malang
Malang, Indonesia

farid@widyagama.ac.id

3rd Gigih Priyandoko

Dept. of Electrical Engineering
Universitas Widyagama Malang
Malang, Indonesia

gigih@widyagama.ac.id;

<https://orcid.org/0000-0001-5498-6248>

4th Nanang Aris Wahyudi

Dept. of Electrical Engineering
Universitas Widyagama Malang
Malang, Indonesia

nanangariswahyudi@gmail.com

Abstract— The Fourth Industrial Revolution (4IR) has transformed manufacturing and maintenance through advanced technologies, including the Internet of Things (IoT), cloud computing, and Decision Support Systems (DSS). In vehicle maintenance, IoT enables real-time data collection from Electronic Control Units to facilitate predictive and periodic maintenance, enhancing reliability and efficiency. This study develops an IoT-based DSS for periodic vehicle maintenance, focusing on data from vehicle sensors to optimize maintenance intervals based on mileage and component conditions. The system was designed using a web server and database infrastructure, integrated with Application Programming Interfaces to enable real-time communication between the vehicle's sensors and a web dashboard. The dashboard displays component statuses and provides maintenance recommendations. Testing involved function and performance evaluations on data communication between an OBD-II scanner, IoT gateway, and web dashboard. The system reads real-time vehicle data effectively, provides maintenance recommendations based on pre-set DSS rules, and updates component status on the web interface. Tests confirmed synchronization between sensor readings and displayed data, with a response delay under 500 ms, meeting real-time system criteria. The IoT-based DSS for vehicle maintenance proved effective in automating maintenance scheduling, providing users with timely and accurate recommendations. This system promotes efficient and proactive vehicle maintenance, aligning with 4IR goals by enhancing operational reliability through real-time data integration and automated decision support.

Keywords— *Internet of Things, Periodic Maintenance, Decision Support System (DSS), Automated Maintenance Scheduling, Real-Time Data Integration*

I. INTRODUCTION

The Fourth Industrial Revolution (4IR) indicates a transformative era in which advanced digital technologies redefine industrial and service processes. Core innovations such as the Internet of Things (IoT), cloud computing, Artificial Intelligence (AI), and next-generation wireless communication systems [1] drive this change, enabling seamless integration between the physical and digital items [2], [3]. These technologies empower sectors like vehicle maintenance with capabilities for real-time data acquisition, predictive analytics, and informed decision-making [4].

In vehicle maintenance, the integration of IoT plays a critical role by enabling real-time monitoring of components,

paving the way for condition-based and predictive maintenance strategies. These strategies involve data from Electronic Control Units (ECUs) [5] and other sensors to enhance maintenance accuracy and reduce downtime [6], [7]. Cloud computing complements IoT by offering scalable computational and storage resources essential for analyzing the large data volumes generated by vehicles [8], [9]. Additionally, Decision Support Systems (DSS), driven by AI, process this data to provide actionable insights, assisting users in decision-making processes critical to maintenance scheduling and fault detection [10], [11].

However, current systems often rely on periodic, pre-set maintenance intervals rather than exploiting the full potential of real-time, condition-based monitoring. This limitation highlights the need for a more advanced IoT-driven DSS approach to optimize maintenance strategies. Furthermore, advancements in wireless communication technologies [12], including the transition from 5G to 6G, introduce opportunities for faster data transmission, improved connectivity, and autonomous decision-making. These capabilities align with the needs of Industry 4.0 and smart cities, where vehicle-to-everything (V2X) communication enhances system efficiency [4], [13].

Optimizing vehicle maintenance through IoT and DSS integration remains challenging despite significant advancements. Existing solutions often fail to fully utilize ECU data for predictive analytics, limiting their effectiveness. Furthermore, communication and system integration limitations resist real-time applications, particularly in resource-constrained environments [5]. Addressing these gaps requires a comprehensive approach that combines cutting-edge IoT frameworks, robust DSS capabilities, and emerging wireless technologies.

This research aims to develop an IoT-based Decision Support System (DSS) for vehicle maintenance. The proposed system utilizes ECU data, such as mileage and sensor readings, enabling periodic and predictive maintenance strategies. The research employs a structured methodology involving system design, API development, and careful testing to ensure the system meets its objectives efficiently.

The paper is organized as follows: Section II presents the Research Methodology, detailing the development process, tools, and testing protocols used to build the IoT-based DSS.

Section III explains the Results and Discussion, covering challenges encountered. Finally, Section IV provides the Conclusion, summing up the overall paper.

II. RESEARCH METHODOLOGY

This research aims to develop a system that can help users in carrying out vehicle maintenance on a periodic basis. The research methodology used can be explained as follows.

A. Method

The research methods used include many steps as follows.

1) *Literature Studies*. The literature study focuses on periodic maintenance methods based on vehicle ECU data [5][14].

2) *Needs Analysis*. This stage is carried out with the aim of better understanding the features and functionalities that must be present in the system.

3) *Data Flow Diagram Planning*. Designing a data flow diagram (DFD) to illustrate the data flow from vehicle sensor data input, scheduling process, and recommendation output.

4) *Web Database Design*. Designing the database for the efficiency of vehicle data management and vehicle maintenance using MySQL with consideration for its ability to handle large-scale data and query speed. phpMyAdmin is used as an interface to facilitate interaction with the database. Entity Relationship (ER) Diagram is used as a design of the data structure in the database.

5) *Web Dashboard Display Design*. The web dashboard is designed simply as a graphical user interface to display information related to vehicle maintenance. The main elements of the web dashboard will be provided so that it can display data in real time and is easy to understand.

6) *Install the Web Server and Setup the Development Environment*. This stage is carried out with the aim of preparing a Web application development environment.

7) *Build APIs for communication between the web server and web user interface (UI)*. The Application Programming Interface (API) needs to be built to support data communication between the server and the web UI. The protocol used is HTTP, with the consideration that HTTP can be easily adopted in environments that have various types of IoT devices because it supports various devices and platforms. HTTP also has acceptable performance in scenarios where security and integration with existing infrastructure are top priorities [15].

8) *Build an API for communication between the web server and the vehicle sensors monitoring system*. This API part is needed to facilitate data communication between the web server and the vehicle sensors monitoring system.

9) *Testing*. Data communication between the vehicle sensor monitoring devices and the web server will be tested via an IoT gateway built with CodeIgniter 4. The data from the vehicle's sensors will be sent to the server, where the system will analyze it and provide maintenance recommendations. The reset feature on the web dashboard is used to update the data calculations.

- *Function Testing*. To ensure each function of the system works according to the specifications.

- *Performance Testing*. Measuring communication and data processing time responses.

10) *Analyze the results*. The analysis is carried out to evaluate the performance of the system in detecting vehicle maintenance schedules and identifying areas that need to be improved. The test results will be analyzed to ensure that the system is functioning as intended and provides accurate recommendations and acceptable response times.

B. Instrumentation and Equipment

1) Research Objects.

- Toyota Avanza 1300 cc G series / Manual Transmission, year 2017.
- DSS for periodic maintenance of a vehicle.

2) Tools.

- ELM327 OBD-II Scanner: This tool is designed to be compatible with a number of vehicles, including a 2017 Toyota Avanza, see Figure 1.
- Redmi Note 13 Android Smartphone: used to run Android Apps;
- OBD Mary scanner Android Apps.: This app is designed to be compatible with a number of OBD scanners installed on a number of vehicles, including a 2017 Toyota Avanza, see Figure 2.
- The IoT gateway: consisting of an Arduino-based Wi-Fi module and a GPRS access electronic module, is used to read sensor data via an OBD-II scanner. The considerations for choosing this device are compatibility, ease of configuration and budget efficiency.

III. RESULTS AND DISCUSSION

The results of this study include testing the system in various aspects that support IoT-based periodic vehicle maintenance. Here are the results of the testing and analysis of each stage.

1) *Literature Studies*. Literature studies show that periodic maintenance of vehicles is carried out based on mileage (km) or a certain time interval (months) [16], [17], as seen in Table I. The test found that the important sensor that can be read from the ECU to support this periodic maintenance is the vehicle mileage (km), see Figure 3., which will be used as an indicator in determining the maintenance status.

2) *Needs Analysis*. This system is designed with the minimum features needed to prove the DSS concept to support IoT-based periodic vehicle maintenance. The main feature is the display of the vehicle's mileage (km) as an indicator of maintenance, as well as several relevant additional parameters, such as water coolant temperature (°C), battery voltage (Volts), and engine rotation (RPM). These features ensure optimal vehicle safety and operation. Vehicle maintenance parameters that need to be monitored and replaced periodically are also displayed, such as engine oil, engine oil filter, fuel filter, air filter, brake fluid, clutch fluid, manual transmission oil, automatic transmission fluid, and rear differential oil.



Fig. 1. ELM327 OBD-II Scanner

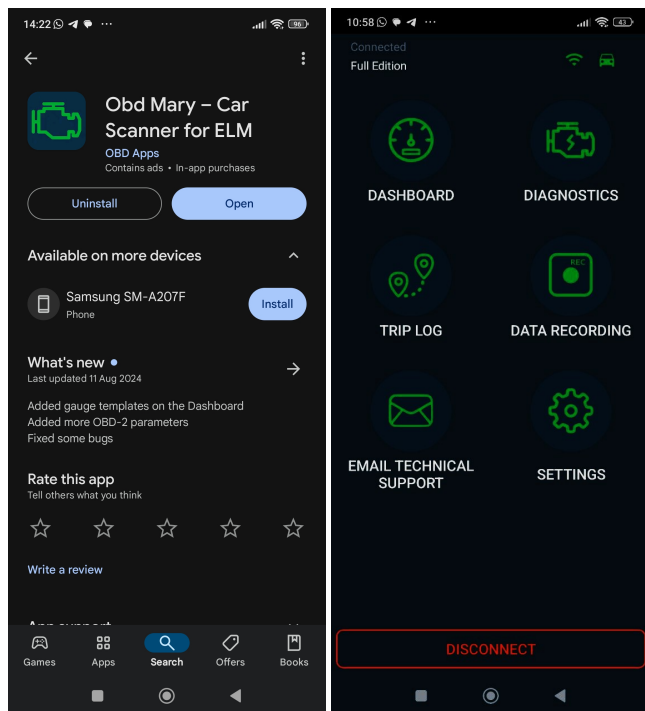


Fig. 2. OBD Mary – Car Scanner for ELM [18].

3) *Data Flow Diagram Planning*. The Data Flow Diagram design (see Figure 4) illustrates the communication between the device that monitors vehicle maintenance parameters and the web server via the IoT gateway. Vehicle parameter data is sent to the web server through HTTP GET requests, which the system processes to display the maintenance status on the web dashboard. APIs are used to securely and efficiently regulate data exchange between the web server and the web UI and between the web server and the device that monitors vehicle sensors, with the Web server storing the data from the sensors and forwarding the relevant data to the web dashboard to be displayed to the

user via the Web-UI. The web server is connected to the MySQL DBMS through the DSS server, which provides decision recommendations to the user regarding periodic vehicle maintenance through the Web UI.

TABLE I. THE VEHICLE PERIODIC MAINTENANCE SCHEDULE [17]

Cara kerja servis berkala / Maintenance operation:
P = Periksa dan perbaiki atau bila perlu ganti / Inspect and correct or replace as necessary; G = Ganti atau tukar / Replace or change;
S = Pelas dan atau setel bila perlu / Check and/or adjust if necessary; L = Lumas dengan gemuk / Lubricate with grease; R = Rotasi ban & balance / Tires rotation & balance;
K = Kencangkan pada momen yang ditentukan / Tightening to the specified torque; (tanda titik) = Ticks ada instruksi servis / No maintenance instructions.

INTERVAL SERVIS BERKALA / SERVICE INTERVAL (Bulan atau pembacaan odometer mana yang lebih dulu) / (Month or odometer reading whichever comes first)	PERIODE SERVIS BERKALA / SERVICE PERIODIC (Bulan / KILOMETER (x 1.000 km) / (Month / KILOMETER (x 1,000 km))	1	2	3	4	5	6	7	8	9	10	11
BAGIAN YANG DISERVIS / MAINTENANCE OPERATIONS	KODE SERVIS BERKALA / MAINTENANCE CODE											
KOMPONEN DASAR MESIN / BASIC ENGINE COMPONENT												
1. Celah katup <<Lihat catatan 1>> / Valve clearance <<See note 1>>		P	P	P	P	P	P	P	P	P	P	P
2. Semut tali kipas / Drive belts		P	P	P	P	P	P	P	P	P	P	P
3. Oli mesin (API SL atau SM, atau ILSAC) / Engine oil (API SL or SM, or ILSAC)		P	G	G	G	G	G	G	G	G	G	G
4. Saringan oli mesin / Engine oil filter		P	G	G	G	G	G	G	G	G	G	G
5. Slang-slang dan persambungan sistem pendinginan dan heater <<lihat catatan 2 dan 3>> / JC cooling and heater system hoses and connections <<See notes 2 and 3>>		P	P	P	P	P	P	P	P	P	P	P
6. Cairan pendingin mesin <<lihat catatan 4 dan 5>> / Engine coolant <<See notes 4 and 5>>		P	P	P	P	P	P	P	P	P	P	P
7. Pipa pipa buang & dukannya / Exhaust pipes & mounting		P	P	P	P	P	P	P	P	P	P	P
SISTEM PENGAPIAN / IGNITION SYSTEM												
8. Busi-busi / Spark plugs - Untuk Rush, Ayra dan Etios / For Rush, Ayra, and Etios - Tipe Iridium / Iridium type - Tipe Lainnya / Other type		P	G	G	G	G	G	G	G	G	G	G
9. Baterai / Battery		P	P	P	P	P	P	P	P	P	P	P



Fig. 3. Some sensors and parameters can be read using OBD Mary.

4) *Web Database Design*. The database is designed to store vehicle data, sensor data, and vehicle maintenance parameters. The function of a database can be explained as follows:

- Store vehicle information data, as well as data that has been successfully read from sensors that contain the latest information on vehicle conditions;
- Provides information for the DSS server to be processed and recommended to the end user.

The database consists of:

- Vehicle data: Contains vehicle type;
- Sensor data: contains several data from the vehicle sensors that are read by the ECU whose data is

updated at all times; namely mileage (km), RPM, battery voltage (Volts), and water coolant temperature (°C);

- Maintenance parameter data and their status: Several vehicle maintenance parameter data include engine oil, engine oil filter, fuel filter, air purifier filter, brake fluid, clutch oil, manual transmission oil, automatic transmission fluid, and rear differential oil. The maintenance status is the condition of the treatment parameter, whether it is "Good" or "Replace".

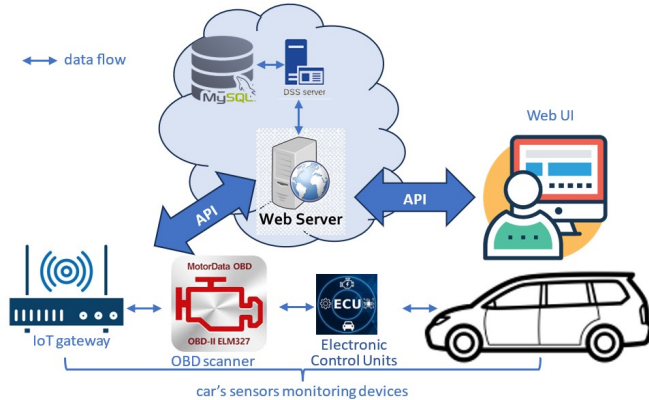


Fig. 4. Illustration of the data flow of DSS-supported periodic vehicle maintenance system

Here are several planned database-structure tables; see Tables II - IV.

TABLE II. VEHICLE MASTER DATA

Name	Type	Information
ID	Int	User ID
Vehicle Name	Varchar	Vehicle Name

TABLE III. SENSOR DATA

Name	Type	Information
ID	Int	Sensor ID
Mileage	Float	Mileage
RPM	Int	Engine rotation speed
Volt	Float	Battery voltage
Temperature	Float	Water Coolant temperature
Created At	DateTime	Time when data comes in
Updated At	DateTime	The last time data was updated

TABLE IV. MAINTENANCE PARAMETERS AND THEIR STATE

Name	Type	Information
ID	Int	Parameter ID
Maintenance Parameter	Varchar	Maintenance Parameter: including "Engine Oil"; "Engine Oil Filter"; "Fuel Filter"; and "Air Cleaning Filter".
Maintenance Distance	Int	Trip (km): 0; 1000; 10000; 20000; 30000; 40000; 50000; 60000; 70000; 80000; 90000; 100000
State	Varchar	Maintenance state: "Replace"; "Good"

The Entity Relationship Diagram of the database design is shown in Figure 5.

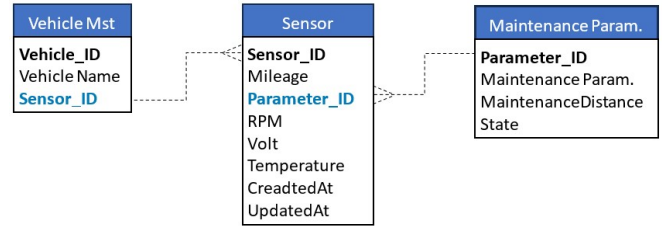


Fig. 5. Proposed Entity Relationship Diagram

This data is then used to support treatment recommendations based on the rules set in the DSS system [17]. The data structure used allows for real-time storage and updating of information.

5) *Web Dashboard Display Design.* The web dashboard is designed using the Bootstrap framework, making it responsive and easily accessible from various devices. This web dashboard displays the maintenance status of vehicle components based on mileage with simple indicators, such as "Replace" or "Good" on each component.

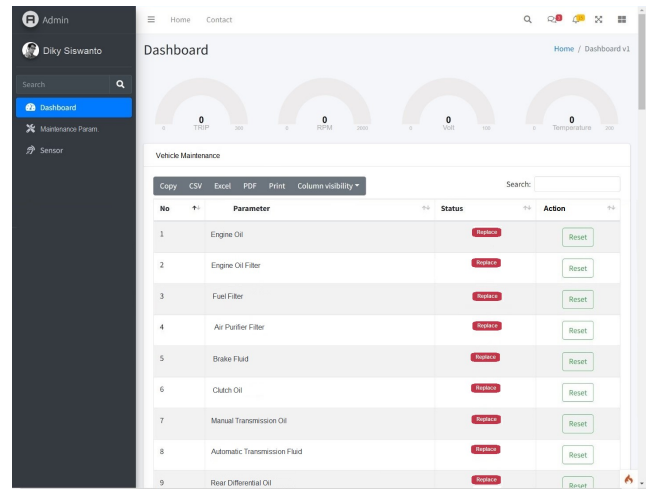


Fig. 6. The web dashboard of the vehicle maintenance monitoring system

Some rules are provided as follows.

- Engine Oil: If the mileage value reaches multiples of 10,000 km, then the status on the table becomes "Replace". If not, then the status is "Good".
- Engine Oil Filter: If the mileage value reaches a multiple of 40,000 km, then the status on the table becomes "Replace". If not, then the status is "Good".
- Brake Fluid: If the mileage value reaches 40,000 km, then the status on the table becomes "Replace". If not, then the status is "Good".
- Fuel Filter: If the mileage value reaches 40,000 km, then the status on the table becomes "Replace". If not, then the status is "Good".
- Air Purifier Filter: If the mileage value reaches 40,000 km, then the status on the table becomes "Replace". If not, then the status is "Good".

6) *Install Web Server and Setup Development Environment.* Apache server, PHP, and Composer installations were performed to create a development environment compatible with CodeIgniter 4. Using the latest versions of PHP and Composer ensures that the system's performance remains stable and secure [19].

7) *Build an API to communicate between the web server and the web UI.* This API is built using the CodeIgniter 4 framework to enable data communication between the Web server and the Web UI. RESTful methods with HTTP GET and POST support real-time data communication needs.

8) *Build an API for communication between the web server and the ECU monitoring device.* The API is also built using the CodeIgniter 4 framework. With this API, the system allows the system to receive real-time data on vehicle sensors read by the ECU, such as mileage (km), engine speed (RPM), battery voltage (Volts), and water coolant temperature (°C).

9) *Testing.* Testing was conducted on sending data from a vehicle monitoring device to a server via an IoT gateway built using CodeIgniter 4 [20]. The DSS server analyzes the received data, then makes some recommendations and then the results are displayed on the web dashboard. The reset button on the web dashboard facilitates updating the calculation data so that it returns to its beginning position.

Testing includes:

Stage 1: Data from the ECU shows an initial mileage of 300,936 km, with the initial trip on the web dashboard being 3 km; see Figures 7–9.



Fig. 7. The odometer shows the number 300.936 km

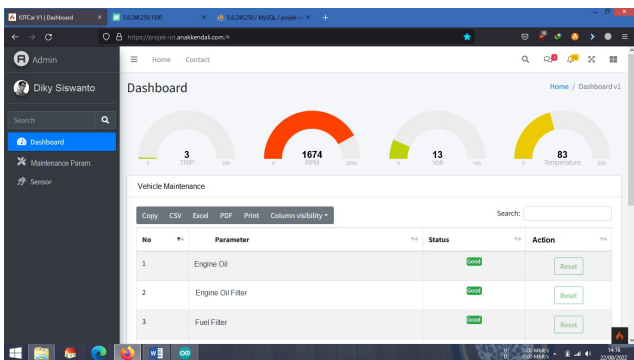


Fig. 8. Web Dashboard

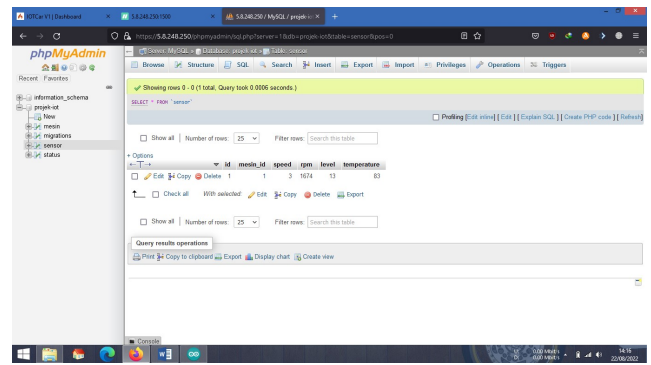


Fig. 9. Web Data Storage Results

Stage 2: The mileage increases to 300,937 km, which increases the trip to 4 km on the web dashboard, showing data synchronization between the sensor and the web dashboard (see Figures 10–12).



Fig. 10. The odometer shows 300.937 km

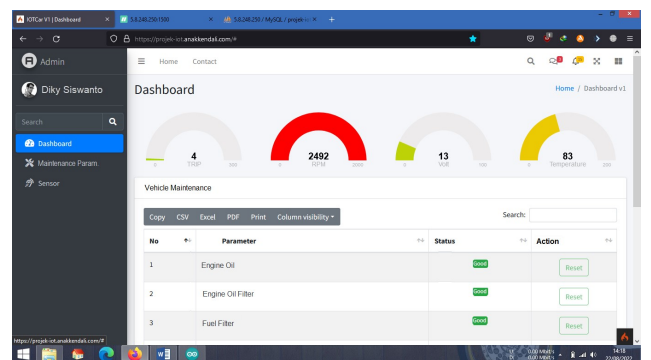


Fig. 11. Web Dashboard

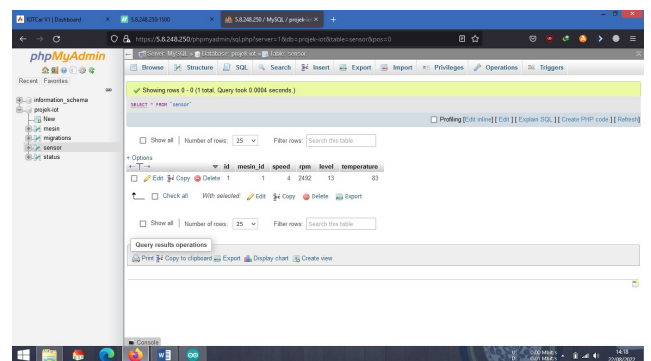


Fig. 12. Results of Data Storage on the Web

The results show that the system effectively reads sensor data in real-time and provides maintenance recommendations based on predefined DSS rules, automating decision-making for vehicle maintenance. Users can monitor component status directly on the web dashboard and receive notifications when a component requires replacement. Testing involves installing an OBD scanner connected to the vehicle's ECU, periodically sending data to the server via an IoT gateway. The server then analyzes this data and generates maintenance recommendations according to DSS rules. For example, (1) Engine Oil: Recommended change every 10,000 km; (2) Oil Filter: Recommended change every 40,000 km; (3) Brake fluid: Recommended change every 40,000 km.

Performance testing shows that the system response is built relatively responsive with a delay time of less than 500 ms, which is considered to meet the criteria of a real-time system [21].

IV. CONCLUSION

This study concludes that the IoT-based vehicle maintenance system developed successfully meets the main goal of reading and monitoring vehicle sensor data in real-time to support periodic maintenance decisions. The system, with a rule-based DSS, can provide maintenance recommendations such as oil and other component changes based on the mileage read through the ECU and automatically update the status of components on the web dashboard. The use of APIs and databases designed to store and process sensor data effectively makes the system responsive with a delay time of less than 500 ms, which meets the criteria of a real-time system. Implementing this system makes it easier for users to manage vehicle maintenance efficiently and proactively and ensure optimal vehicle safety and performance through component replacement notifications.

ACKNOWLEDGMENT

We want to express our sincere gratitude to the DIRJEN-DIKTI, KEMENDIKBUD-RISTEK, the Republic of Indonesia for their invaluable support and funding under the Research Grant with contract No. 073/E5/P6.02.00.PT/2022, on the scheme of PTUPT. This grant has provided resources and encouragement for completing this research project.

REFERENCES

- [1] D. Siswanto, L. Zhang, K. Navaie, and Deepak, "Weighted Sum Throughput Maximization in Heterogeneous OFDMA Networks," *IEEE 83rd Veh. Technol. Conf. VTC2016-Spring*, 2016, [Online]. Available: <https://ieeexplore.ieee.org/document/7504531/>
- [2] McKinsey & Company, "What are Industry 4.0, the Fourth Industrial Revolution, and 4IR?," McKinsey & Company. Accessed: Oct. 20, 2024. [Online]. Available: <https://www.mckinsey.com/featured-insights/mckinsey-explainers/what-are-industry-4-0-the-fourth-industrial-revolution-and-4ir>
- [3] M. Majid *et al.*, "Applications of Wireless Sensor Networks and Internet of Things Frameworks in the Industry Revolution 4.0: A Systematic Literature Review," *Sensors*, vol. 22, no. 6, 2022, doi: 10.3390/s22062087.
- [4] A. Z. Yonis and A. Nawaf, "Investigation of Evolving Multiple Access Technologies for 5G Wireless System," in *2022 8th International Engineering Conference on Sustainable Technology and Development (IEC)*, Erbil, Iraq: IEEE, 2022, pp. 118–122. doi: 10.1109/IEC54822.2022.9807471.
- [5] Z. Fox, "Detecting anomalies using real-time ECU processing activity," US10289404B2, 2019 [Online]. Available: <https://patents.google.com/patent/US10289404B2/en>
- [6] S. N. Swamy and S. R. Kota, "An empirical study on system level aspects of Internet of Things (IoT)," *IEEE Access*, vol. 8, pp. 188082–188134, 2020, doi: 10.1109/ACCESS.2020.3029847.
- [7] S. Kumar, P. Tiwari, and M. Zymbler, "Internet of Things is a revolutionary approach for future technology enhancement: a review," *J. Big Data*, vol. 6, no. 1, p. 111, 2019, doi: 10.1186/s40537-019-0268-2.
- [8] Peter Mell and T. Grance, *The NIST Definition of Cloud Computing: Recommendations of the National Institute of Standards and Technology*. United States, 2011, pp. 1–3. [Online]. Available: <https://nvlpubs.nist.gov/nistpubs/legacy/sp/nistspecialpublication800-145.pdf>
- [9] H. A. Khattak, H. Farman, B. Jan, and I. U. Din, "Toward Integrating Vehicular Clouds with IoT for Smart City Services," *IEEE Netw.*, vol. 33, no. 2, pp. 65–71, 2019, doi: 10.1109/MNET.2019.1800236.
- [10] S. Reddy, J. Fox, and M. P. Purohit, "Artificial intelligence-enabled healthcare delivery," *J. R. Soc. Med.*, vol. 112, no. 1, pp. 22–28, 2019, doi: <https://doi.org/10.1177/0141076818815510>.
- [11] Z. Zhai, J. F. Martínez, V. Beltran, and N. L. Martínez, "Decision support systems for agriculture 4.0: Survey and challenges," *Comput. Electron. Agric.*, vol. 170, p. 105256, 2020, doi: <https://doi.org/10.1016/j.compag.2020.105256>.
- [12] D. Siswanto, L. Zhang, and K. Navaie, "Spectrum splitting-based cognitive interference management in two-tier LTE networks," in *Wireless Communications Systems (ISWCS), 2014 11th International Symposium on*, Aug. 2014, pp. 613–617.
- [13] A. Z. Yonis, H. A. Dweig, and A. K. Tareed, "Comprehensive analysis of IEEE 802.11ah for Wireless Communication Networks," in *2021 IEEE Integrated STEM Education Conference (ISEC)*, 2021, pp. 28–31. doi: 10.1109/ISEC52395.2021.9763924.
- [14] S. Navneeth, K. P. Prithvil, N. R. Sri Hari, R. Thushar, and M. Rajeswari, "On-Board Diagnostics and Driver Profiling," in *2020 5th International Conference on Computing, Communication and Security (ICCCS)*, 2020, pp. 1–6. doi: 10.1109/ICCCS49678.2020.9277449.
- [15] M. Amundsen, *RESTful Web API Patterns and Practices Cookbook*. O'Reilly Media, Inc., 2022. [Online]. Available: O'Reilly Media, Inc.
- [16] J. Crawford and R. Kimball, *Beginner's Auto Maintenance & Repair*. Arizona, USA: Mirex Marketing, 2014. [Online]. Available: <http://crawfordsautoservice.com/>
- [17] Technical Service Division, *Buku Warranty & Service Berkala/ Warranty & Service Booklet: Bensin/ Gasoline*, 2nd ed. Jakarta: Toyota-Astra Motor, 2017.
- [18] antipovapps@gmail.com, "OBD Mary – Car Scanner for ELM." Google Play store, 2024. [Online]. Available: <https://obdscanner.net/>
- [19] The PHP Group, "php: A popular general-purpose scripting language that is especially suited to web development. Fast, flexible and pragmatic, PHP powers everything from your blog to the most popular websites in the world.," www.php.net. Accessed: Nov. 06, 2024. [Online]. Available: <https://www.php.net/>
- [20] O. W. Purbo, "A Systematic Analysis: Website Development using Codeigniter and Laravel Framework," *Enrich. J. Manag.*, vol. 12, no. 1, pp. 1008–1014, 2021, doi: <https://doi.org/10.35335/enrichment.v12i1.346>.
- [21] M. Suznjevic and J. Saldana, "Delay Limits for Real-Time Services," *IETF Draft*, 2016, [Online]. Available: https://www.researchgate.net/publication/304674431_Delay_Limits_for_Real-Time_Services