 ISSN 2579-7298

Internet of Things (IoT) Platform Using Web Services-Based The Laravel Framework in Wind Turbines

Yudhis Thiro Kabul Yunior

Airport Electrical Engineering, Surabaya Aviation Polytechnic,Jl.Jemur Andayani I No.73 Surabaya, 60236,Indonesia
yudhis.kabul@poltekbagsby.ac.id
* corresponding author

ARTICLE INFO

Article history
Received
Revised
Accepted

Keywords
Internet of Things
laravel
Web Service
Wind Turbine

ABSTRACT

The increasing need for sustainable electrical energy has driven the development of various renewable energy sources, one of which is wind power plants (wind turbines). The development of the Internet of Things (IoT) is a solution for monitoring and managing wind power plants remotely, in real time so that it is more efficient and centralized. The integration method on the Internet of Things (IoT) platform is to build a sensor system installed on the wind turbine output so that it can send operational data in real time. The background of this study will design and implement a power monitoring system on a wind turbine using an Internet of Things-based web service platform. The development of an Internet of Things-based power monitoring system on web services will be built using the Laravel framework which will be integrated with the power output of the wind power plant. This platform is expected to simplify the process of monitoring, controlling, and analyzing wind turbine performance in real time, so that it can increase operational efficiency and support the development of sustainable renewable energy.

1. Introduction

The increasing need for sustainable electrical energy has encouraged the development of various renewable energy sources, one of which is wind power plants (wind turbines). Wind turbines utilize the kinetic energy of the wind to produce electrical energy in a clean and environmentally friendly manner. However, challenges in managing and monitoring wind turbine performance in real-time are obstacles to optimizing the efficiency of wind power plants, especially in remote areas far from control centers.

Internet of Things (IoT) technology is emerging as a solution to monitor and manage wind power plants more efficiently and centrally. With IoT, various sensors installed on wind turbines can transmit real-time operational data, such as wind speed, turbine condition, power output, and other parameters, which are then processed to provide valuable insight in decision making. However, to integrate this data effectively, a platform is needed that is reliable and easily accessible by various devices.

Laravel, as a modern PHP-based web framework, offers solutions for developing robust and scalable web services platforms. With various features that support the development of service-based web applications, such as flexible routing, authentication, and API integration, Laravel can be the right choice for building an IoT platform that manages data from wind turbines efficiently.

Based on the above background, this research will design and implement an IoT-based web services platform using the Laravel framework for wind power generation. This platform is expected to simplify the process of monitoring, controlling and analyzing wind turbine performance in real-time, so as to increase operational efficiency and support the development of sustainable renewable energy.

2. Method

The research output is to design and implement an Internet of Things (IoT) based web services platform using the Laravel framework for monitoring and data management for wind power plants (wind turbines). The methodology used in this research includes several stages of system design.

A. Embedded System Design

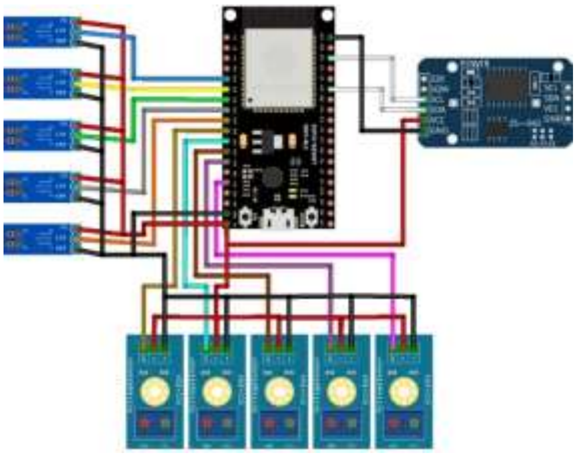

Fig 1. Design of Embedded System

At this stage, an embedded system architecture is designed which consists of several main components, namely IoT sensor devices in the form of voltage and current sensors installed on the wind turbine to measure parameters such as wind speed, power output, temperature and turbine condition. Data from sensors will be sent to the server using communication protocols such as MQTT or HTTP. The Web Server will be designed using the Laravel framework which functions to receive, store and manage data from IoT devices. Laravel is used to build APIs that enable efficient data exchange between servers and IoT devices. Input data from current and voltage sensors will be sent to a database management system which stores all operational data from the wind turbine, such as sensor measurement results, operational status, as well as event or error logs.

B. Data Base Management System Design

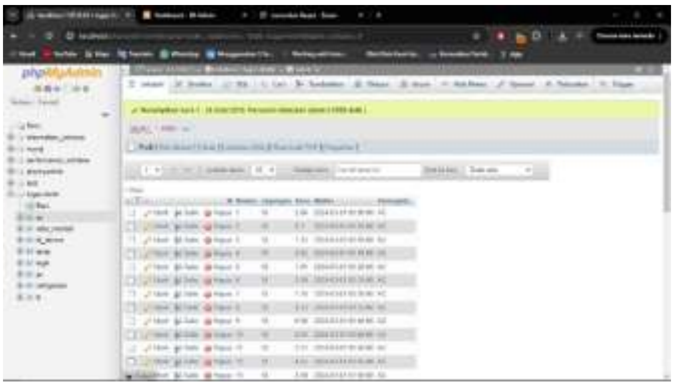

Fig 2. Database Management Desgin

In designing this database management system, input data from current and voltage sensors will be sent to a database management system which stores all operational data from the wind turbine, such as sensor measurement results, operational status, as well as event or error logs. Sensor data. This will then be integrated into the User Experience (UX) backend system and displayed on the

Web Services User Interface (UI) dashboard. The system uses a relational database (MySQL) to store data sent from sensors. The table structure is designed to accommodate various parameters such as wind speed, temperature, and electrical power output. Database capacity testing shows that the system is capable of storing large amounts of data without experiencing significant performance degradation.

3. Results and Discussion

A. Development Web Service Dashboard

Fig 3. Dashboard Web Service IoT Wind Turbine

The display above is a web services dashboard interface that allows users to monitor data in real-time, carry out analysis, and control the wind turbine if necessary. A web-based dashboard was developed to display wind turbine operational data in the form of graphics, reports and statistics that are easy for users to understand. This dashboard also provides a notification feature if there are abnormal conditions or errors in the turbine. Data is displayed in the form of graphs and tables, providing clear and informative visualization. Users can see trends in wind speed, electrical power output, and receive notifications if there are abnormal conditions such as a decrease in turbine efficiency. Test results show that this interface is easy to use and the data displayed corresponds to the data sent from the sensor.

B. Backend Using Laravel Testing

Fig 4 Code for Laravel Framework Testing

At this stage, the development of the web services platform begins by building an API using the Laravel framework. Laravel is used to handle routing, authentication, and processing of data sent from IoT devices. Features such as API authentication and user management have also been developed to maintain data security. IoT devices connected to the wind turbine will be integrated with the web services platform via API. Data from sensors is sent to the Laravel server using the MQTT or HTTP protocol, then the data is stored in the database for users to access.

C. Testing API System

Fig 5 API Testing Code

The testing stage is carried out to ensure that all system components function properly. Testing was carried out on several aspects. API testing is carried out to measure API performance in receiving and processing data from IoT devices in real-time. Testing is also carried out at the System Security stage. This test is carried out to ensure that the system can protect data sent from IoT devices, as well as avoid unauthorized access. Next is testing the dashboard functionality. This test is carried out to ensure that data is displayed accurately and that users can interact with the system intuitively. API performance testing is carried out by sending data from sensors continuously over a certain period of time. The results show that the Laravel API is able to receive and process data in real-time without experiencing bottlenecks. The average response time of the API to receive data is less than 1 second, which shows its efficiency in handling large amounts of data.

4. Conclusion

The final stage of this research is to draw conclusions based on the results obtained from system design, implementation and testing. In addition, recommendations for further development were also prepared, such as improving system scalability, integration with solar power plants, or the use of AI technology to predict future wind turbine performance.

This methodology is expected to produce a web services platform that can facilitate monitoring and management of wind turbines efficiently and in real-time, as well as making a real contribution to the development of sustainable renewable energy.

The results of the performance analysis show several recommendations that this web services platform is able to facilitate wind turbine monitoring efficiently and in real-time. Some of the metrics analyzed to make recommendations include:

A. Monitoring Efficiency

This platform allows wind turbine monitoring to be carried out continuously and without pause. Users can monitor important parameters such as wind speed, temperature and power output from remote locations, without the need to carry out direct inspections at the turbine site. This increases operational efficiency as response times to problems can be accelerated.

B. Ease of Use

The developed dashboard provides intuitive visualization, with graphs and notifications that help users understand wind turbine conditions easily. The responsive interface design ensures that users can access the platform from a variety of devices, including smartphones and tablets.

C. Reduced Downtime

With real-time notifications, users can immediately find out if there is a problem with the wind turbine. This enables quick action to correct problems and reduces downtime, which directly results in increased electricity production efficiency.

References

[1] Ghani, A., et al. (2018). "IoT-based energy management in smart grid: Applications, challenges, and future directions." *Renewable and Sustainable Energy Reviews*, 91, 90-108.

[2] Kounev, V., et al. (2019). "IoT-based monitoring and control system for energy-efficient smart grid infrastructure." *Journal of Electrical and Computer Engineering*, 2019.

[3] C. Baier, J-P. Katoen, Principles of Model Checking, MIT Press, 2008.

[4] Kashyap, M., et al. (2020). "IoT-enabled solar power generation monitoring and energy forecasting system." *International Journal of Energy Research*, 44(12), 9782-9795.

[5] Gupta, V., & Singh, A. (2019). "IoT-based solar power monitoring system using cloud platform." *International Journal of Engineering and Advanced Technology*, 8(6), 4540-4544.

[6] Li, W., et al. (2021). "Smart energy management in IoT-based microgrids for renewable energy production." *IEEE Transactions on Industrial Informatics*, 17(8), 5657-5668.

[7] Haider, H., et al. (2020). "Real-time energy management in microgrid systems using IoT technology." *International Journal of Smart Grid and Clean Energy*, 9(4), 738-745.

[8] Khan, M. A., et al. (2021). "IoT-enabled renewable energy systems: Challenges and future directions." *Sustainable Cities and Society*, 70, 102925.

[9] Wadhwa, S., et al. (2019). "IoT-based real-time energy monitoring in smart homes with renewable energy integration." *International Journal of Smart Grid and Clean Energy*, 8(2), 100-108.

[10] B. Meyer, Applying "Design by Contract", Computer 25(10) (1992) 40–51. DOI: https://doi.org/10.1109/2.161279

[11] Reddy, G. R., et al. (2020). "IoT-based remote monitoring and control of solar power plants." *International Journal of Renewable Energy Research*, 10(3), 1034-1042.

[12] Al-Ali, A. R., et al. (2017). "IoT-based smart grid power monitoring system." *International Journal of Smart Grid*, 1(1), 11-22.

[13] Stauffer, Matt. (2019). *Laravel: Up & Running: A Framework for Building Modern PHP Apps*. 2nd ed. O'Reilly Media. ISBN: 978-1492041215

[14] Bean, Martin. (2015). *Laravel 5 Essentials*. Packt Publishing. ISBN: 978-1785283017

[15] Pecoraro, Christopher John. (2017). *Mastering Laravel*. Packt Publishing.ISBN: 978-1785882814

[16] Sinha, Sanjib. (2017). *Laravel RESTful Web Services*. Apress. ISBN: 978-1484226940

[17] Mitchell, Lorna Jane. (2013). *PHP Web Services: APIs for the Modern Web*. O'Reilly Media. ISBN: 978-1449356569