POWER ANALYSIS OF A 100 WATT MICROHYDRO POWER GENERATOR USING AN INTERNET OF THINGS (IoT) WEB SERVICE BASED ON THE CODE IGNITER FRAMEWORK

Yudhis Thiro Kabul Yunior[1,*]

[1] *Politeknik Penerbangan Surabaya*

ABSTRACT

Small-scale Microhydro power plants (100 Watt) require a real-time and accurate power monitoring system to improve efficiency and maintenance. This research aims to develop an Internet of Things (IoT)-based electrical power analysis system by utilizing a Web Service based on the CodeIgniter Framework to process and display data online. The system consists of sensor modules (voltage and current) using ZMPT101B and ACS712, an ESP32 microcontroller for data transmission, and a CodeIgniter backend that provides a RESTful API for data storage and processing. Power (P), voltage (V), current (I), and energy (kWh) data are displayed on a web dashboard with graphic visualization using Chart.js. The research method uses a Research and Development (R&D) approach with stages of needs analysis, system design, implementation, and testing. The test results show that the system is able to monitor power with 95% accuracy compared to digital multimeter measurements, and has a data transmission latency of <2 seconds. This solution can be applied to small-scale Microhydro power plants for IoT-based monitoring with low cost and high scalability.

Keywords: *: Internet of Things (IoT) ,Web Service, CodeIgniter, Microhydro 100 Watt*

1. INTRODUCTION

The development of Internet of Things (IoT) technology has brought significant transformations to various sectors, including power generation. IoT enables real-time data collection, monitoring, and analysis, thereby improving system efficiency and reliability. One type of power plant currently being developed is Micro Hydro, a small-scale hydroelectric power plant with a capacity of up to 100 watts. This plant is suitable for remote areas not yet connected to the main grid. However, to ensure optimal performance, an accurate and accessible real-time power monitoring and analysis system is required.

Currently, power measurements at Micro Hydro plants are still often performed manually or using conventional devices that are not integrated with digital systems. This leads to delays in problem identification, a lack of historical data for analysis, and difficulties in making maintenance decisions. Therefore, an IoT-based solution is needed that can automatically monitor and analyze power and present data through an easily accessible web interface.

Web services are an effective approach for integrating IoT devices with web-based monitoring systems. Using the CodeIgniter framework, web service development can be carried out efficiently because it provides a clean structure, strong security, and ease of developing RESTful APIs for communication between IoT devices and servers.

This research aims to develop an electrical power analysis system for the Micro Hydro 100 Watt power plant using a CodeIgniter-based IoT web service. This system is expected to provide real-time information on voltage, current, power, and plant efficiency, as well as store historical data for further analysis. This allows operators or users to monitor plant performance more effectively and perform preventative maintenance if necessary.

It is hoped that the results of this research will contribute to improving the efficiency of small-scale power plants, particularly Micro Hydro, and serve as a reference for the development of IoT systems in the renewable energy sector.

2. METHOD

The research method that will be used in this research is Research and Development (R&D), namely a systematic approach to developing new products, systems, or technologies through a series of stages of research, design, testing, and refinement.

The steps in the Research and Development (R&D) approach using the prototyping method in this study include:

• Needs Analysis
• System Design
• IoT and Web Service Implementation
• Testing and Evaluation
Development Stages

a. Requirements Analysis

The functional requirements identification stage (sensor readings, data storage, visualization) and IoT architecture design are as follows: Sensor → ESP32 → Web Service (CodeIgniter) → Database → Dashboard

b. System Design

The system design flowchart is as follows:
1. The sensor reads voltage and current.
2. Data is sent via WiFi to the Web Service (CodeIgniter).
3. The server processes and stores the data.
4. The data is displayed on the web dashboard.
The database design consists of:
The power_data table with columns for id, voltage, current, power, energy, and timestamp.

c. IoT and Web Service Implementation

ESP32 firmware with programming using the Arduino IDE to send data to the API.
CodeIgniter backend with REST API creation to receive and store data. Dashboard frontend with data visualization using Chart.js or Highcharts.

d. System Testing

Sensor testing with sensor calibration using a multimeter. API testing using the Postman tool for endpoint testing to obtain response time and error rate metrics. Accuracy testing by comparing system data with standard measuring instruments.

6. Data Analysis

Quantitative analysis by calculating the error rate of sensor readings and evaluating web service performance (latency, throughput). Qualitative analysis by testing user satisfaction.

3. RESULT AND DISCUSSION

a) Design System

Figure 1. Work Principle of Micro hydro power Plant

The attached image depicts a Micro Hydro Power Plant (PLTMH) system combined with an Internet of Things (IoT) architecture for real-time data monitoring and tracking. The overall working principle of this system focuses on converting the kinetic energy of water into electricity, followed by data collection, processing, and visualization through the IoT layer.

b)Microhydro Power Plant

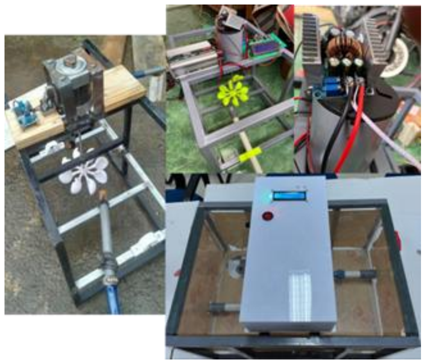

Figure 2. 100 Watt MicroHydro Power Plant

Technically, the 100-watt Micro-hydro system is designed to be simple and modular for easy DIY assembly, as seen in the prototype's component set, which includes:

(1) A screw turbine made of plastic or lightweight metal with spiral blades for high efficiency at low heads (1-5 meters);

(2) A permanent magnet generator (PMG) mounted on the turbine shaft for mechanical-to-electrical conversion;
(3) A wooden or metal support frame for stability;
(4) A PVC or HDPE pipe system for water distribution, equipped with a small weir to divert the flow without building a large dam;
(5) Supporting electronics such as a rectifier, voltage regulator, and LCD display for real-time power output monitoring; and
(6) A container or control box to protect the components from environmental elements. Inductors and capacitors in the electronic circuit stabilize the DC output current, while a simulated DC motor is used in initial testing to mimic the turbine's rotation. The system's efficiency reaches 60-70% under optimal conditions, with minimal head loss in the pipes thanks to the adjusted diameter..

This prototype demonstrates high potential for off-grid applications. At a water flow rate of 0.05 m^3/s and a head of 2 meters, the output power is stable at 80-100 watts, sufficient to illuminate 5-10 LED lights (5-10 watts each) or operate a small pump. Field testing confirmed that seasonal flow fluctuations can be mitigated by adding a storage battery (e.g., a 24 VDC bank) and a simple inverter for AC conversion if needed. However, major challenges include water sedimentation, which can clog the turbine blades, requiring intake filters and regular maintenance every 3-6 months. Furthermore, efficiency decreases at heads below 1 meter, emphasizing the importance of selecting a site with a natural gradient.

c) Design Web Services

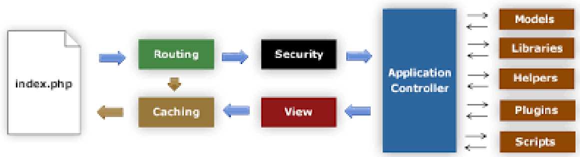

Figure 3 Web Services Architecture

The image above shows the basic architectural flow of the PHP CodeIgniter (CI) framework for building web service applications. CodeIgniter adopts a modified Model-View-Controller (MVC) design pattern for efficiency, where user requests are processed sequentially through various layers before generating a response. This diagram shows a simple flowchart from the entry point file to the supporting components, with a focus on how CI handles routing, security, caching, and integration with the main controller.

The MVC architecture in CodeIgniter adopts the MVC (Model-View-Controller) pattern, which separates application logic into three components:

•RESTful API development (for IoT communication).

•Integration with MySQL/PostgreSQL databases (sensor data storage).

•Monitoring dashboard creation (as in the Mikro Hydro research).

Implementation Example:

```
class Sensor extends CI_Controller {
public function get_data() {
$data = $this->db->get('sensor_readings')->result();
echo json_encode($data);
}
}
```

d) Design Js Chart

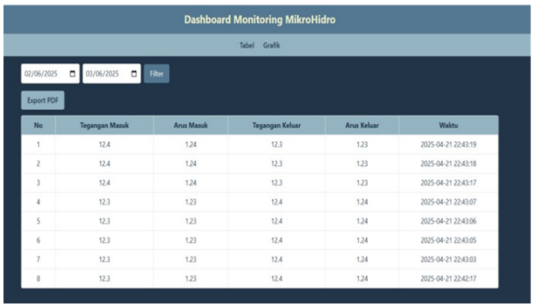

Figure 4. Dashboard Web Services

The image above displays a web-based MicroHydro Monitoring Dashboard with a responsive and interactive tabular interface, designed to monitor the performance of Micro-Hydro Power Plants (PLTMH) in real-time or historically. This dashboard is built using CodeIgniter technology integrated with IoT and databases, presenting sensory data from the PLTMH through an API or direct connection to a microcontroller (NodeMCU).

This dashboard is a reliable, scalable, and user-friendly web-based monitoring solution for small-scale PLTMHs (100 watts). With CodeIgniter, IoT, and database architecture, the system is capable of storing thousands of records per day, presenting data visually, and supporting data-driven decision-making for preventative maintenance and renewable energy optimization in remote locations.

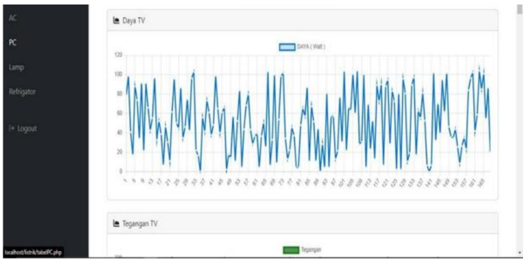

Figure 5. Chart.js Web Services

The image above displays a web dashboard interface for monitoring the power consumption of several electronic devices generated by a Micro-Hydroelectric Power Plant (PLTMH) system using Chart.js as a data visualization library. The libraries used include:

```
new Chart(ctx, {
  type: 'line',
  data: {
    labels: ['08:00', '09:00', ..., '22:00'], // Waktu
(sumbu X)
    datasets: [{
      label: 'DAYA ( Watt )',
      data: [45, 78, 92, 55, 88, ...], // Nilai daya real-
time
      borderColor: '#007bff',
      backgroundColor: 'rgba(0, 123, 255, 0.1)',
      fill: true,
      tension: 0.4
    }]
  },
  options: {
    responsive: true,
    scales: {
      y: { min: 0, max: 120, title: { display: true,
text: 'Watt' } },
      x: { title: { display: true, text: 'Waktu' } }
    },
    plugins: {
      title: { display: true, text: 'Daya TV' },
      tooltip: { mode: 'index', intersect: false }
    }
  }
```

Chart.js on this dashboard successfully visualizes two key parameters: Power (Watts) and Voltage (Volts). The chart data demonstrates the stability of the micro hydropower plant's output despite fluctuating loads. The advantages of Chart.js implementation include real-time, responsiveness, and user-friendliness. Chart.js is also fully integrated with the CodeIgniter framework, IoT, and MySQL database management. The development of the Chart.js dashboard can support decision-making such as overload detection, regulator efficiency evaluation, and maintenance planning.

REFERENCES

[1] Atzori, L., Iera, A., & Morabito, G. (2010). The Internet o Things: A survey. Computer Networks.

[2] Gungor, V. C., et al. (2011). Smart Grid Technologies: Communication Technologies and Standards. IEEE.

[3] Kumar, A., et al. (2019). IoT-based Monitoring of Hydroelectric Power Plants. Springer.

[4] CodeIgniter Documentation. (2023). CodeIgniter 4 User Guide. https://codeigniter.com/

[5] Rahardian, D. (2020). Pemrograman Web dengan Framework CodeIgniter. Informatika.

[6] Firmansyah, A., et al. (2021). IoT-Based Monitoring System Using CodeIgniter Framework. IJCCS.

[7] Borg, W. R., & Gall, M. D. (1983). Educational Research: An Introduction.

[8] YTK Yunior , Internet of Things (IoT) Platform Using Web Services-Based The Laravel Framework in Wind Turbines, 2025, International Journal of Artificial Intelligence Research. DOI: https://doi.org/10.29099/ijair.v9i1.1.1468

[9] Irfansyah, A., Yunior, Y. T. K., & Harianto, B. B. (2023). Internet of Things (IoT) Based Trash Can Monitoring System Using Web Services with the CodeIgniter Framework in the Surabaya Aviation Polytechnic Environment. In *Proceeding of International Conference of Advanced Transportation, Engineering, and Applied Social Science* (Vol. 2, No. 1, pp. 1050-1053).

[10] SHS Hariadi, YTK Yunior - Journal of Research in Social Science and Humanities, 2025. DOI: https://doi.org/10.47679/jrssh.v5i1.348

[11] Putra, K. E., Yunior, Y. T. K., & Sukma, M. M. (2024, December). IMPLEMENTATION OF A SEPIC CONVERTER IN AN OFF-GRID SOLAR POWER PLANT (PLTS) USING FUZZY LOGIC METHOD WITH A HAIWELL SCADA-BASED MONITORING SYSTEM. In *Proceeding of International Conference of Advanced Transportation, Engineering, and Applied Social Science* (Vol. 3, No. 1, pp. 232-237).

[12] Nanditta, Meyra, Yudhis Thiro Kabul Yunior, and Meita Maharani Sukma. "DESIGN AND DEVELOPMENT OF CONTROL MONITORING FOR REAL-TIME CLOCK (RTC) IMPLEMENTATION IN A SOLAR CELL TRACKER USING NAIVE BAYES METHOD BASED ON WEB SERVICE." *Proceeding of International Conference of Advanced Transportation, Engineering, and Applied Social Science*. Vol. 3. No. 1. 2024.

[13] Efrial, M. A., Yunior, Y. T. K., & Julaihah, S. (2024, December). IMPLEMENTATION OF IOT IN THE RADIATOR COOLANT CONTROL AND MONITORING PROTOTYPE ON GENSET USING ESP 32 BASED NODEMCU SMARTPHONES. In *Proceeding of International Conference of Advanced Transportation, Engineering, and Applied Social Science* (Vol. 3, No. 1).