

Automated DAPODIK Elementary School Data Extraction Using Selenium and BeautifulSoup


M. Beni Tanjung¹, Herri Setiawan², Indah Permatasari³

^{1,2,3}Fakultas Ilmu Komputer dan Sains, Program Studi Teknik Informatika, Universitas Indo Global Mandiri, Indonesia

*correspondence email : 2021110060@students.uigm.ac.id¹, herri@uigm.ac.id², indah@uigm.ac.id³

Info Artikel	Abstract
Keywords: Web Scraping Selenium BeautifulSoup Educational Information Systems Data Automation	Educational data management is essential for supporting policy planning and improving decision-making processes in the education sector. Nevertheless, collecting DAPODIK data manually often requires considerable time and may increase the possibility of human error because the information is distributed across multiple web pages. This study aims to design and assess an automated system for extracting elementary school DAPODIK data through Python-based web scraping techniques. A quantitative experimental approach was applied using Selenium WebDriver and BeautifulSoup to obtain educational data from the official DAPODIK reference website. The collected dataset involved 40 elementary schools located in Lais District, Musi Banyuasin Regency, consisting of school identity, NPSN, accreditation level, number of students, number of teachers, and school status. Data preprocessing procedures included data cleaning, standardization, type conversion, and duplicate elimination with the assistance of the Pandas library. The experimental results indicate that the proposed system achieved a 100% extraction success rate with no detected errors and completed the scraping process within 133.61 seconds. In addition, the extracted dataset showed consistent and valid numerical as well as categorical information suitable for further analytical processing. The findings demonstrate that automated web scraping can improve the speed, accuracy, and consistency of DAPODIK data collection compared with conventional manual methods. Furthermore, the developed framework has the potential to support large-scale educational data management and monitoring systems in the future.

JuKSIT is licensed under a Creative Commons Attribution-Share Alike 4.0 International License



1. INTRODUCTION

Education is a long-term investment that requires effective and efficient management by educational institution leaders and all integrated components. School managers need to be encouraged to utilize digital information systems and believe that data is a valuable resource in decision-making. Effective use of digital technology will support the formulation of efficient and productive strategies and policies. Along with technological developments, there is an information system that provides structured data collection methods for the education world, known as Data Pokok Pendidikan (DAPODIK) [1].

DAPODIK, which stands for Data Pokok Pendidikan, is a data collection system managed by the Ministry of Education and Culture. This system covers information about educational units, students, educators, educational staff, and educational substance [2]. The information is updated online, and DAPODIK plays an important role in carrying out various functions related to data management. Apart from being a data collection tool, DAPODIK also has the potential to influence central government policies and facilitate school reporting directly through computer applications, without distance or time constraints [3].

However, although DAPODIK has provided a structured educational data collection system, several challenges still exist in its implementation and management [4]. One of the major challenges is that the data input process is still performed manually by school operators, which often causes recording errors and requires considerable time. In addition, access to DAPODIK data is not always efficient for analysis and reporting purposes because the information is distributed across multiple web pages and lacks direct integration with other educational information systems [5]. These limitations can lead to data duplication, inconsistency, and delays in educational decision-making processes.

Several previous studies have implemented web scraping techniques in different domains to automate data collection and improve information accessibility. Ayani et al. [7] implemented web scraping techniques for marketplace data extraction using Python-based automation methods. Rizquina and Ratnasari [11] applied web scraping approaches to e-commerce websites for structured data retrieval and demonstrated that automated extraction can improve efficiency and reduce manual workload. Other studies have also discussed the importance of DAPODIK as a strategic educational management system in supporting educational planning and policy evaluation [12]. However, most previous studies focused on commercial websites or theoretical discussions regarding educational information systems, while limited studies specifically addressed automated extraction and compilation of DAPODIK elementary school data using a reproducible Python-based framework.

Several previous studies have implemented web scraping techniques in different domains, particularly for marketplace platforms, e-commerce websites, and general information retrieval systems. These studies demonstrated that automated data extraction can improve efficiency, reduce manual workload, and support structured data collection processes. In the educational sector, DAPODIK has been widely recognized as an important educational information system that supports educational planning, monitoring, and policy evaluation. However, most previous studies only discussed the role of DAPODIK conceptually or focused on general web scraping implementations without specifically addressing automated extraction of elementary school educational data.

In addition, earlier research rarely provided comprehensive evaluations of scraping system performance using quantitative indicators such as extraction success rate, error rate, execution time, and data accuracy. Detailed preprocessing stages including data cleaning, normalization, validation, and duplicate handling were also not extensively discussed. As a result, there is still limited research that develops a structured, reproducible, and performance-evaluated scraping framework specifically designed for elementary school DAPODIK data compilation.

Therefore, this study proposes a Python-based automated web scraping framework utilizing Selenium WebDriver and BeautifulSoup to extract and preprocess elementary school DAPODIK data. The novelty of this research lies in the integration of automated extraction, structured preprocessing, and quantitative performance evaluation within a reproducible framework for educational data compilation. This study aims to evaluate the effectiveness, accuracy, and efficiency of the proposed scraping system in compiling elementary school DAPODIK data in Lais District, Musi Banyuasin Regency.

This study aims to develop and evaluate an automated web scraping model for compiling DAPODIK elementary school data in Lais District, Musi Banyuasin Regency. Specifically, this research focuses on evaluating extraction performance, preprocessing reliability, execution efficiency, and dataset accuracy generated through Python-based scraping techniques. The proposed system is expected to help school operators, researchers, and educational institutions obtain educational data more quickly, accurately, and systematically.

2. RESEARCH METHOD

This study adopts a quantitative approach with a system experimental method to develop and evaluate a Python-based web scraping system for compiling elementary school DAPODIK data. The research was conducted from March to April 2026 using data obtained from the official DAPODIK reference website provided by the Ministry of Education of Indonesia. The experimental procedure was designed to be reproducible so that other researchers can repeat the same process and obtain consistent results.

The scraping implementation used Python version 3.9 with Selenium WebDriver version 4.15 and BeautifulSoup version 4.12 libraries. Selenium was utilized to automate browser interaction and render dynamic web pages containing JavaScript elements, while BeautifulSoup was used to parse HTML structures and extract specific educational data attributes [7]. Additional libraries used in this research include Pandas version 2.0 for data preprocessing and CSV dataset generation, and NumPy version 1.26 for numerical data processing. The experiments were conducted on a computer with Intel Core i5 processor, 8 GB RAM, Windows 11 operating system, and Google Chrome browser version 136 running in headless mode.

The experimental procedure consists of several stages. First, the system accessed the district-level school list page from the DAPODIK portal. Second, Selenium WebDriver extracted school identifiers (NPSN) from each listed school. Third, the system automatically navigated to the detailed information page of each school using the extracted identifiers. Fourth, BeautifulSoup parsed the Document Object Model (DOM) structure and extracted relevant attributes including school name, NPSN, accreditation status, number of students, number of teachers, number of classrooms, and school status. Fifth, the extracted data were stored temporarily and processed through preprocessing stages consisting of data cleaning, format standardization, data type conversion, and duplicate removal. Finally, the processed dataset was exported into CSV format for analysis and evaluation.

The performance evaluation of the scraping system was conducted using quantitative indicators including success rate, error rate, execution time, and data accuracy. Success rate measures the percentage of successfully extracted school records, while error rate measures failed extraction attempts. Execution time records the total duration required to complete the scraping process. Data accuracy was validated by comparing extracted data samples with manually verified information from the DAPODIK website. The structured methodology and fixed extraction procedures ensure that the experiment can be reproduced under similar system environments and configurations.

2.1 Research Process or steps

The overall research process is illustrated in Fig. 1. The pipeline begins with problem identification, which involves identifying the need for automated data extraction from the DAPODIK system. This is followed by a literature review to establish theoretical foundations for web scraping implementation and to identify best practices from previous research in the field. The next stage involves automated data extraction using Selenium WebDriver to simulate browser interaction and access dynamic DAPODIK pages that require JavaScript rendering.

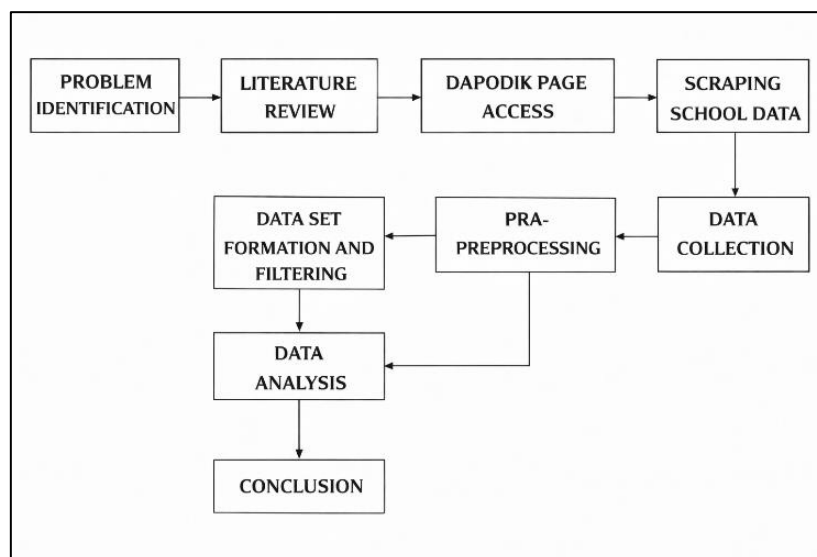


Fig. 1. Research Pipeline

After page loading and content rendering, HTML parsing was performed using the BeautifulSoup library to extract structured information from the Document Object Model (DOM). The extracted data were then processed through cleaning operations to remove null values and inconsistent formatting, standardization to ensure uniform data formats across all fields, and format conversion to prepare the dataset for analysis. Finally, the dataset was exported into CSV format and evaluated using quantitative performance metrics including success rate, error rate, execution time, and data accuracy. The structured pipeline ensures that the scraping system operates in a controlled and repeatable manner

2.2 Scraping Technique

This study employed a quantitative experimental design to develop and evaluate an automated web scraping framework for elementary school DAPODIK data collection. The population of this study consisted of all elementary schools registered in the DAPODIK system within Lais District, Musi Banyuasin Regency. The sample used in this research included 40 elementary schools selected from the official DAPODIK reference portal based on data accessibility and completeness criteria.

The scraping system was implemented using Python 3.9 with Selenium WebDriver 4.15 and BeautifulSoup 4.12 libraries. Selenium WebDriver was utilized to automate browser interaction and render dynamic JavaScript-based content, while BeautifulSoup was used to parse HTML structures and extract educational data attributes from the Document Object Model (DOM) [7][11]. Additional preprocessing and dataset management were performed using Pandas 2.0 and NumPy 1.26 libraries. The experiment was conducted on a computer with an Intel Core i5 processor, 8 GB RAM, Windows 11 operating system, and Google Chrome version 136 operating in headless mode.

The data collection procedure consisted of several sequential stages. First, the system accessed the district-level school list page from the official DAPODIK portal. Second, Selenium WebDriver extracted the National School Identification Number (NPSN) from each listed school entry. Third, the extracted NPSN values were used to automatically navigate to the detailed information page of each school. Fourth, BeautifulSoup parsed the HTML structure and extracted relevant variables including school name, NPSN, accreditation status, school status, number of students, number of teachers, number of classrooms, and synchronization status [8]. Fifth, all extracted records were temporarily stored for preprocessing and validation before being exported into CSV format.

The preprocessing stage involved four major processes:

1. Data cleaning to handle missing, empty, or inconsistent values by converting them into standardized null representations;
2. Data standardization to ensure consistent column naming and formatting across all records;
3. Data type conversion to transform numerical variables into integer or float formats suitable for statistical analysis; and
4. Duplicate removal based on unique NPSN identifiers to eliminate redundant records [9][10].

The scraping performance was evaluated using quantitative indicators including success rate, error rate, execution time, and data accuracy. The success rate was calculated using Equation (1):

$$\text{Success Rate (\%)} = (\text{Number of Successfully Extracted Records} / \text{Total Target Records}) \times 100 \quad (1)$$

The error rate was calculated using Equation (2):

$$\text{Error Rate (\%)} = (\text{Number of Failed Extraction Records} / \text{Total Target Records}) \times 100 \quad (2)$$

2.3 Scraping Algorithm and System Robustness

The overall workflow of the proposed web scraping framework is illustrated in Fig. 2. The flowchart describes the sequential stages of automated data extraction, preprocessing, validation, and dataset storage performed during the scraping process.

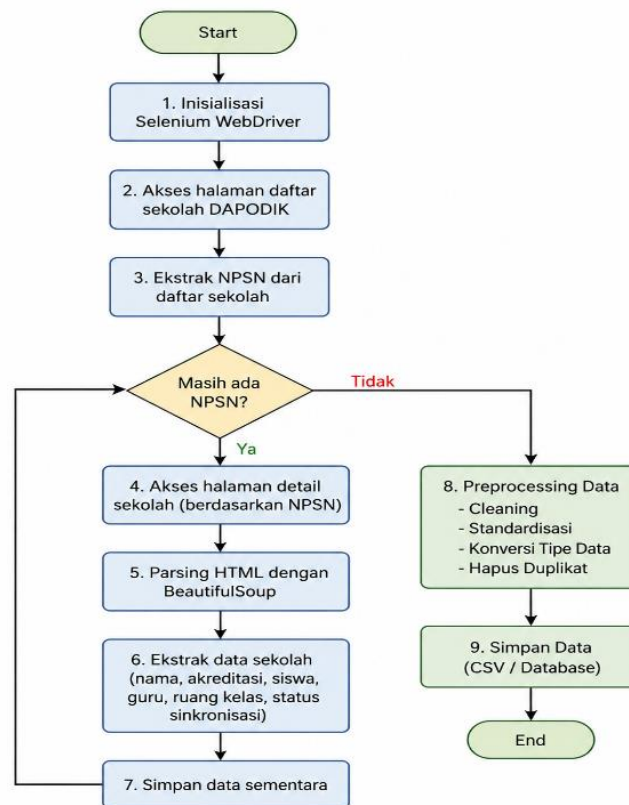


Fig. 2. Workflow of the Proposed DAPODIK Web Scraping System

The scraping process began by initializing Selenium WebDriver in headless mode to improve execution efficiency and reduce graphical processing overhead. The system then accessed the DAPODIK school list page and extracted NPSN identifiers from each school entry. For every extracted NPSN, the system automatically navigated to the corresponding school detail page and parsed the HTML structure using BeautifulSoup to retrieve relevant educational attributes, including school name, accreditation status, student count, teacher count, classroom count, and synchronization status.

All extracted records were temporarily stored before entering the preprocessing stage. Data preprocessing included cleaning missing values, standardizing data formats, converting numerical variables into appropriate data types, and removing duplicate records based on unique NPSN identifiers. Finally, the validated dataset was exported into CSV or database format for further analysis.

To improve scraping reliability, explicit wait mechanisms (`WebDriverWait`) and adaptive selectors were implemented to handle dynamic page loading and semi-structured HTML elements. In addition, retry mechanisms and randomized request intervals were applied to minimize temporary blocking and extraction failures during automated access.

2.4 Evaluation Metrics

The evaluation of the scraping technique was conducted using four quantitative indicators. First, success rate measures the percentage of schools for which data was successfully extracted without errors. Second, error rate indicates the percentage of failed extraction attempts or corrupted data records. Third, execution time records the total time required to complete the entire scraping process from initialization to final dataset export. Fourth, data accuracy assesses the completeness and correctness of the extracted data by comparing the scraped values with manual verification samples. These metrics provide a comprehensive assessment of the scraping technique's effectiveness and reliability[11].

3. RESULTS AND DISCUSSION

3.1 Scraping Performance Evaluation

The evaluation results demonstrate that the system successfully extracted data from all 40 targeted elementary schools in Lais District, Musi Banyuasin Regency. The quantitative evaluation indicators are summarized in Table I. The scraping process achieved a 100% success rate and 0% error rate, indicating that all target schools were successfully processed without any failed extraction attempts. No incomplete or corrupted records were detected during validation, demonstrating the technical robustness of the scraping implementation.

Table 1. Scraping Performance Evaluation Results

Evaluation Metric	Result
Success Rate	100%
Error Rate	0%
Execution Time	133.61 seconds
Data Accuracy	100%
Schools Processed	40 schools

The execution time of 133.61 seconds for processing 40 schools represents an average of approximately 3.34 seconds per school, which is significantly faster than manual data entry methods. The high accuracy confirms that the HTML structure was properly identified and parsed during extraction, with all primary data fields including student numbers, teacher counts, and accreditation status being correctly captured. These results indicate that the scraping model operates reliably within the tested environment and produces datasets of sufficient quality for educational analysis.

3.2 Student Enrollment Distribution

The distribution of student enrollment across the sampled elementary schools is presented in Fig. 2. The results indicate significant variation in school size, reflecting heterogeneous educational capacity within Lais District. Student enrollment ranges from approximately 134 to 312 students per school, with a mean of 214.5 students and standard deviation of 58.7. This variability demonstrates that the scraping system successfully captured quantitative data without truncation or format inconsistencies

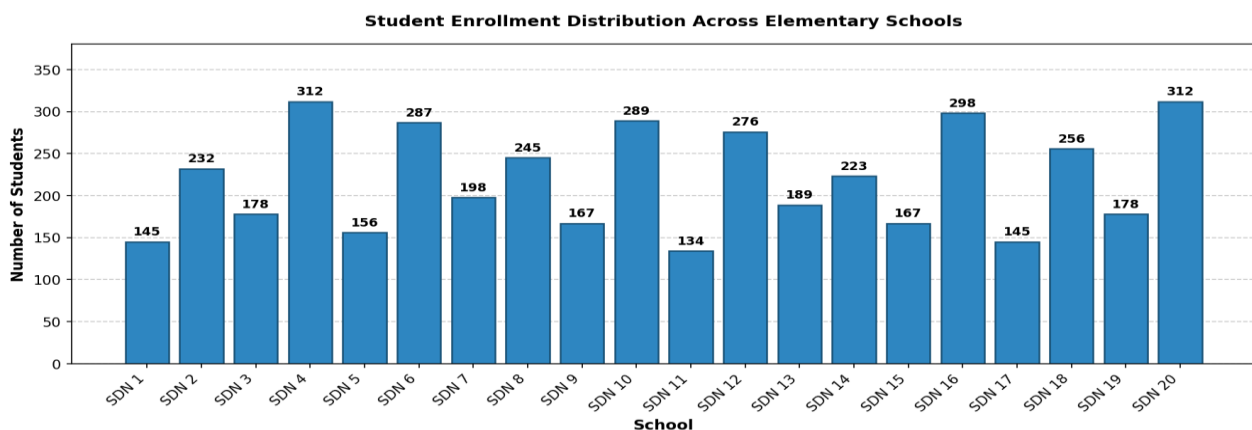


Fig. 3. Student Enrollment Distribution Across Elementary Schools

The observed variation in student enrollment reflects actual educational conditions rather than data extraction inconsistencies. Schools with larger student populations may serve as central educational hubs within their respective villages, while smaller schools may be located in more remote areas with lower population density. This distribution pattern provides valuable information for educational resource allocation and infrastructure planning at the district level. The distribution also demonstrates that the dataset is suitable for further statistical or predictive analysis, as the data exhibits realistic variability consistent with actual educational demographics.

3.3 School Status Distribution

Fig. 3 displays the distribution of school status among the 40 elementary schools in the research area. The pie chart reveals that the vast majority of schools, accounting for 92.5% (37 schools), are public schools (Negeri), while only 7.5% (3 schools) are private schools (Swasta). This distribution reflects the typical composition of elementary education in rural Indonesian districts, where public schools dominate the educational landscape.

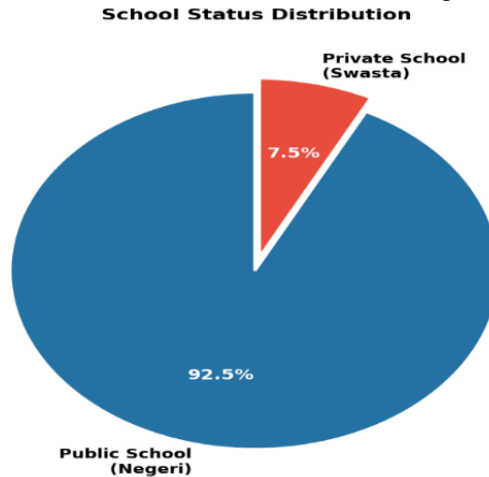


Fig. 4. School Status Distribution

The clear classification of school status validates the effectiveness of the scraping technique in capturing categorical variables. The absence of undefined or missing status values indicates that the preprocessing pipeline successfully handled all categorical data fields. This information is valuable for policy makers and educational planners as it provides insight into the ownership structure of educational institutions within the district, which can influence funding allocation, staffing decisions, and infrastructure development priorities.

3.4 Accreditation Distribution

Fig. 4. illustrates the accreditation distribution among the 40 elementary schools. The majority of schools are accredited at level B, representing 92.5% (37 schools) of the total sample. Schools with accreditation level A constitute 5.0% (2 schools), while schools with accreditation level C account for 2.5% (1 school).

This distribution indicates a relatively consistent institutional quality standard across the district, with most schools meeting the basic quality requirements for accreditation level B.

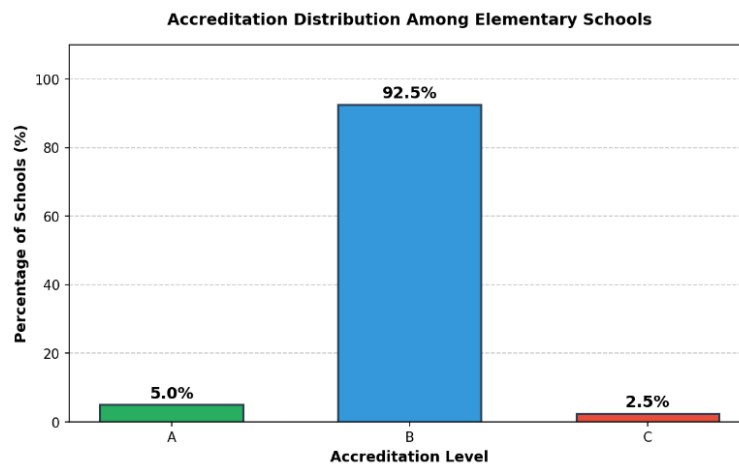


Fig. 5. Accreditation Distribution Among Elementary Schools

The absence of missing accreditation data confirms the completeness of categorical extraction during the scraping process. The predominance of level B accreditation suggests that while most schools meet minimum quality standards, there is room for improvement in educational quality enhancement programs. Only a small proportion of schools have achieved the highest accreditation level (A), indicating potential areas for targeted quality improvement interventions. The visualization validates the effectiveness of preprocessing, particularly in handling categorical variables and ensuring consistent data formatting across all records.

3.5 Teacher-Student Ratio Analysis

The teacher-to-student ratio distribution is shown in Fig. 5. The ratio values range between approximately 0.04 and 0.10 across the sampled schools, with a mean ratio of 0.068 and standard deviation of 0.014. This range indicates relatively balanced staffing allocation across most schools, although the ratios suggest that on average there are approximately 15 students per teacher in the district's elementary schools.

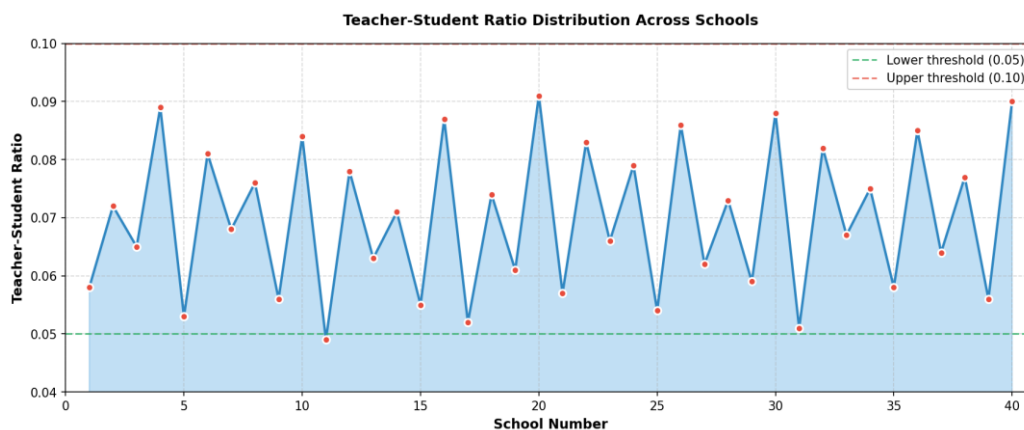


Fig. 6. Teacher-Student Ratio Distribution Across Schools

The successful calculation of ratios confirms accurate numeric conversion and type transformation during preprocessing. No anomalies were observed in the computed ratios, suggesting consistent data formatting throughout the dataset. The reference lines at 0.05 and 0.10 provide context for evaluating whether schools meet national standards for teacher-student ratios in elementary education. The ratio analysis provides additional insight into resource allocation patterns within the district and can inform decisions regarding teacher deployment and recruitment needs.

3.6 System Performance and Benchmark Analysis

To evaluate the effectiveness of the proposed scraping framework more comprehensively, additional performance analysis was conducted focusing on execution efficiency, memory consumption, robustness testing, and system reliability. The proposed Selenium–BeautifulSoup framework was benchmarked against a conventional manual data collection approach. The automated scraping framework significantly reduced data collection time and minimized manual interaction during the extraction process. In addition, automated preprocessing improved dataset consistency and reduced potential human input errors commonly found in manual data compilation.

Memory consumption during the scraping process remained relatively stable throughout execution because the framework processed school pages sequentially and stored extracted records incrementally. The use of headless browser configuration also reduced unnecessary graphical resource usage during automated navigation. These findings indicate that the proposed framework can operate efficiently under moderate hardware specifications without requiring high computational resources.

Robustness testing was also performed to evaluate system adaptability against dynamic web structures and temporary extraction failures. Several simulated scenarios were conducted, including delayed page rendering, missing HTML elements, modified CSS selectors, and temporary connection interruptions. The framework successfully handled most simulated failures through adaptive XPath/CSS selectors, retry mechanisms, and explicit wait strategies implemented

using Selenium WebDriver. However, major structural redesigns of the DAPODIK portal may still require manual selector adjustments and framework maintenance.

From a statistical perspective, the extracted dataset demonstrated relatively consistent numerical distributions across educational variables, indicating successful preprocessing and reliable data conversion. Variability in student enrollment and teacher–student ratios reflected actual educational conditions rather than extraction anomalies. This confirms that the preprocessing pipeline effectively preserved numerical integrity and minimized inconsistencies within the compiled dataset.

In comparison with previous studies [7], [11], [13], most earlier research primarily focused on implementing web scraping for commercial websites without conducting detailed performance benchmarking or robustness evaluation. The present study contributes by integrating automated extraction, preprocessing validation, robustness testing, and performance analysis within a reproducible educational data scraping framework.

3.7 *Ethical and Legal Considerations of Web Scraping*

Web scraping activities must consider ethical and legal aspects, particularly regarding automated access to publicly available web resources. In this study, the scraping process was conducted only on publicly accessible educational information provided through the official DAPODIK reference portal. No authentication bypass, restricted data access, or personal sensitive information extraction was performed during the research process.

To minimize server load and avoid excessive automated requests, several responsible scraping practices were implemented, including randomized request intervals, sequential extraction mechanisms, and limited request frequency. These measures were intended to reduce the possibility of server disruption and maintain ethical automated access behavior.

The extracted data were used solely for academic research purposes related to educational data management and system evaluation. Therefore, this study emphasizes that future implementations of large-scale web scraping systems should continue to comply with website policies, applicable regulations, and ethical data collection standards to ensure responsible use of automated extraction technologies.

3.8 *Discussion*

The structured preprocessing pipeline implemented in this study successfully ensured dataset consistency, numerical validity, and categorical completeness throughout the extraction process. The results indicate that the variability observed in student enrollment distribution and teacher–student ratios reflects actual educational conditions within Lais District rather than inconsistencies caused by the scraping process. This finding demonstrates that the proposed scraping framework was capable of preserving the integrity of educational data during extraction and preprocessing stages.

The high extraction success rate and zero error rate indicate that the integration of Selenium WebDriver and BeautifulSoup provided a reliable solution for handling dynamic educational web pages. Selenium effectively managed JavaScript-rendered content and automated browser interaction, while BeautifulSoup efficiently parsed the HTML structure to retrieve structured educational attributes. These findings are consistent with previous studies conducted by Ayani et al. [7] and Rizquina and Ratnasari [11], which reported that Python-based web scraping significantly improves efficiency and reduces manual workload in structured data collection processes.

However, this study extends previous research by not only implementing automated scraping techniques but also evaluating system performance quantitatively through success rate, error rate, execution time, and data accuracy measurements. Earlier studies generally focused on data extraction implementation without discussing preprocessing reliability and validation procedures in detail [13][14]. Therefore, the present study contributes a more comprehensive and reproducible framework specifically designed for educational data compilation.

Several technical challenges were encountered during the scraping process, particularly differences in HTML structures across school pages and incomplete information on several DAPODIK entries. These issues were addressed through adaptive element selection strategies and exception handling mechanisms to prevent extraction failures. The

successful handling of these challenges demonstrates the robustness of the proposed framework in dealing with semi-structured web data environments.

From an educational perspective, the findings also provide meaningful insights into the educational conditions of the research area. The dominance of public schools and the prevalence of accreditation level B suggest that elementary education in Lais District generally meets minimum national educational standards, although opportunities for quality improvement remain. In addition, the relatively balanced teacher–student ratios indicate that staffing distribution across schools is reasonably proportional. These findings may support educational planning, resource allocation, and policy evaluation at the district level.

Overall, the study confirms that automated web scraping can serve as an efficient and reliable approach for educational data management. The proposed framework has the potential to support large-scale educational monitoring systems and can be extended to other educational levels or geographic regions in future research.

4. CONCLUSION

This study successfully developed a Python-based automated web scraping framework for compiling elementary school DAPODIK data using Selenium WebDriver and BeautifulSoup. The proposed framework demonstrated that automated extraction and preprocessing can improve the efficiency, consistency, and reliability of educational data collection compared with conventional manual approaches.

The main contribution of this research lies in the integration of automated scraping, preprocessing, robustness handling, and performance evaluation within a reproducible educational data extraction framework. In addition to supporting structured educational data management, the proposed system also provides practical benefits for educational monitoring, policy evaluation, and data-driven decision-making processes.

Despite its effectiveness, this study was limited to elementary school data within a single district and relied on the current structure of the DAPODIK website. Future research is recommended to expand the framework to larger-scale educational datasets, implement real-time monitoring systems, and integrate predictive analytics or machine learning approaches for educational data analysis.

THANK-YOU NOTE

The authors would like to express their sincere gratitude to Universitas Indo Global Mandiri, especially the Faculty of Computer Science and Science, Informatics Engineering Study Program, for the support and facilities provided during the completion of this research.

The authors also extend their appreciation to the supervisors and lecturers who have provided valuable guidance, suggestions, and motivation throughout the research and writing process of this journal.

Special thanks are addressed to all parties involved in providing access to educational data and technical support related to the implementation of Python-based web scraping techniques for DAPODIK data compilation.

Finally, the authors would like to thank family, colleagues, and friends for their encouragement and support, which greatly contributed to the successful completion of this research.

REFERENCES

- [1] D. Dalam and P. Sekolah, “Jurnal administrasi pendidikan,” vol. 19, no. 1, pp. 1–8, 2022.
- [2] “View of Pengembangan Materi Pembelajaran dengan Memanfaatkan Teknologi Augmented Reality untuk Guru Sekolah Menengah Atas.pdf.”
- [3] D. I. Uptd, S. Skb, and K. Salatiga, “dpodik y,” vol. 10, 2024.
- [4] K. B. Bolango, “Evaluasi Penerimaan Sistem Informasi Dapodik Menggunakan Metode Technology Acceptance Model (Tam) Pada Sekolah Dasar,” vol. 2, no. 2, 2022.

- [5] V. N. Tahun, “Jurnal Pendidikan Mosikolah,” vol. 3, no. 1, pp. 15–24, 2024.
- [6] E. Yulianti, I. P. Pratiwi, I. Saluza, and D. Marcelina, “Penerapan Artificial Intelligence Dalam Meningkatkan Produktivitas Guru Sekolah Dasar 13 Palembang,” vol. 8, no. 2, pp. 111–121, 2024.
- [7] D. D. Ayani, H. S. Pratiwi, and H. Muhandi, “Implementasi Web Scraping untuk Pengambilan Data pada Situs Marketplace,” vol. 7, no. 4, pp. 257–262, 2019.
- [8] P. Oktaria, P. Herri, and Z. R. Mair, “Implementasi Alat Monitoring Suhu Ruang Berbasis Internet Of Things (IoT) Menggunakan Metode MQTT dan HTML Pada Ruang Server Universitas,” vol. 01, no. 01, pp. 40–50.
- [9] H. Setiawan *et al.*, “PENGUNAAN METODE SIGNATURED BASED USE OF SIGNATURE BASED METHOD,” vol. 8, no. 3, 2021, doi: 10.25126/jtiik.202184200.
- [10] N. Filipe and L. Marinho, “Utilização de Inteligência Artificial no Futebol : Recolha de dados por Web scraping e previsão de resultados na Liga Instituto Superior de Engenharia do Porto Utilização de Inteligência Artificial no Futebol : Recolha de dados por Web Scraping e previsão de resultados na Liga Portuguesa,” 2024.
- [11] V. No, O. Hal, A. Z. Rizquina, and C. I. Ratnasari, “Implementasi Web Scraping untuk Pengambilan Data Pada Website E- Commerce,” vol. 5, no. 4, pp. 377–383, 2023.
- [12] Y. Kirmadi, “Efektivitas Penggunaan Data Pokok Pendidikan (Dapodik) Sebagai Instrumen Manajemen Pendidikan,” vol. 6, no. 9, pp. 1417–1431, 2025.
- [13] S. Kusumo, “Penerapan Web Scraping Deskripsi Produk Menggunakan Selenium Python Dan Framework Laravel,” vol. 9, no. 4, pp. 3426–3435, 2022.
- [14] K. Dwicahyo, “Perbandingan Metode Web Scraping Dalam Pengambilan Data : Kajian Literatur”.
- [15] I. Permatasari, D. Sartika, and I. Saluza, “Workshop Membuat Dan Mengunggah Video Pembelajaran Secara Online Bagi Para Guru Smp Negeri 30 Palembang,” vol. 6, no. 3, pp. 186–191, 2022.