



Improving Data Completeness in SINTA Publication Scraping Using an Iterative Method

Muhammad Arfah Asis¹, St. Hajrah Mansyur², Nia Kurniati³

^{1,3}Information Engineering, Faculty of Computer Science, Universitas Muslim Indonesia, Makassar, Indonesia

²Informatics Systems, Faculty of Computer Science, Universitas Muslim Indonesia, Makassar, Indonesia

muh.arfah.asis@umi.ac.id

Abstract

The structure of publication data on lecturer profiles in SINTA, particularly those indexed by SCOPUS, often results in data duplication and missing records. This issue arises because articles are distributed by year across multiple pages, making standard single-pass scraping methods unable to guarantee data completeness. This study aims to develop and evaluate the effectiveness of an iterative scraping method in improving the accuracy of publication data retrieval from SINTA. The proposed method involves a series of ten experimental trials, in which the results of single-pass scraping are compared with those of iterative scraping. The evaluated parameters include the level of data completeness and the number of iterations required to achieve optimal results. The findings indicate that single-pass scraping captures only an average of 70.7% of publications in the first iteration, with frequent occurrences of duplicated and missing data. In contrast, the iterative scraping method consistently achieves 100% publication retrieval across all trials, although it requires a varying number of iterations ranging from four to eleven. Therefore, it can be concluded that iterative scraping is a more reliable approach for ensuring the completeness and accuracy of publication data. Although this approach demands greater computational resources than standard methods, it is well suited for large-scale bibliometric studies, institutional evaluations, and more comprehensive monitoring of research trends.

Keywords: author profile; iterative scraping; publication data; SINTA; web scraping

How to Cite: M. A. Asis, S. H. Mansyur, and N. Kurniati, "Improving Data Completeness in SINTA Publication Scraping Using an Iterative Method", *J. RESTI (Rekayasa Sist. Teknol. Inf.)*, vol. 10, no. 3, pp. 548 - 554, May 2026.
Permalink/DOI: <https://doi.org/10.29207/resti.v10i3.6952>

Received: July 13, 2025

Accepted: May 2, 2026

Available Online: May 23, 2026

This is an open-access article under the [CC BY 4.0 License](#)

Published by [Ikatan Ahli Informatika Indonesia](#)

1. Introduction

SINTA (Science and Technology Index) is a national scientific indexing system developed by the Ministry of Education, Culture, Research, and Technology of the Republic of Indonesia [1]. This platform serves as a centralized database that facilitates the monitoring and evaluation of scientific publication performance, citations, research collaborations, and researcher profiles from various educational and research institutions in Indonesia [2-5]. The data provided by SINTA is utilized not only by academic and educational institutions but also as a reference in decision-making related to accreditation, policy formulation, and research funding.

With the growing demand for publication data across institutions [6-8], many parties have begun developing systems capable of automatically accessing information from SINTA. One of the most commonly used methods

is web scraping, a technique for extracting information from web pages by simulating user interactions in a programmatic manner [9]. This method works by extracting HTML elements from web pages and converting them into structured data [10-12]. This approach is particularly useful for collecting large-scale data, such as the complete list of lecturer publications from various institutions, which is not readily available through the standard user interface. For example, web scraping has also been widely used to extract publication data and researcher profiles from Google Scholar [13-20], including author names, institutional affiliations, citation counts, and links to individual articles. This indicates that web scraping is not only relevant in the context of SINTA but also applicable to other platforms that provide scientific publication information.

Although SINTA previously provided an official API, the service is currently no longer available. As a result,

accounts owned by a particular university cannot retrieve publication data of lecturers from other institutions, and even regional coordinator accounts do not have full access across regions. Therefore, web scraping of SINTA publication pages remains a relevant alternative [21], especially for collecting data on a national or inter-institutional scale.

Nevertheless, the web scraping process on SINTA is not without challenges. Preliminary observations reveal that SINTA's publication pages often display inconsistent data in terms of both order and content. The same page may present different entries upon repeated access, which means that a one-time data retrieval process cannot guarantee the completeness and accuracy of the information. This phenomenon introduces risks of duplication and data loss, particularly in large-scale data collection. In practice, duplication may occur when the same article appears on multiple pages due to shifting pagination or inconsistent ordering within the same publication year. Such duplicates can inflate publication counts and bias bibliometric evaluations, making their detection and removal a critical step in the scraping process.

Several previous studies have implemented web scraping techniques on SINTA, primarily focusing on extracting article data [22] or journal metadata [1], [2], [23], [24]. However, these studies have not specifically addressed issues related to data instability or its impact on scraping outcomes. Furthermore, no studies have explicitly evaluated whether the scraping process can produce complete and duplicate-free datasets. These conditions delineate a scientific research gap because instability-induced duplication and missing records have not been operationalized or empirically verified, which threatens the validity and reproducibility of bibliometric indicators derived from SINTA.

To address these challenges, this study proposes an iterative web scraping approach, which involves repeated scraping of each targeted page until all data is comprehensively retrieved. Each scraping result is systematically compared to detect and eliminate duplicate entries. With this approach, the system is expected to produce datasets that are accurate, consistent, and complete.

The aim of this study is to establish a reliable data retrieval framework for SINTA that can overcome issues of data instability and duplication. This approach brings two main benefits. First, it ensures the integrity of bibliometric analyses derived from SINTA data. Second, it provides a methodological foundation that can be applied to other dynamic academic databases facing similar challenges. The main contribution of this research lies in the design and implementation of an iterative scraping method that differs from conventional single-pass approaches by repeatedly retrieving and comparing data until a stable and complete dataset is obtained. To the best of our knowledge, this is the first study that introduces and evaluates such an iterative scraping strategy in the context of SINTA, thereby

offering a novel methodological solution for large-scale publication data collection within a national context.

2. Methods

This study adopts an experimental approach to develop and evaluate an iterative web scraping method for extracting publication data from lecturer profile pages on the SINTA platform. Each author profile in SINTA contains three categories of publications: SCOPUS, GARUDA, and Google Scholar. This study specifically focuses on extracting data from the SCOPUS category, as the data in this section is generally more structured, stable, and is considered a key indicator in assessing a lecturer's research performance.

The scraping process was implemented using the PHP programming language, utilizing the Simple HTML DOM Parser library, which enables efficient parsing of HTML elements [14]. The iterative web scraping approach was chosen to address page instability issues that often result in data duplication or loss when scraping is performed only once.

The method consists of four main stages: page structure analysis, scraping module design, iterative scraping process, and validation and evaluation of the scraping results.

2.1 Page Structure Analysis

The first step involved analyzing the HTML structure of the SCOPUS publication section on lecturer profile pages in SINTA. Key elements such as article titles, journal names, author names, year of publication, and journal quartiles were identified based on consistent HTML tags and class structures. This identification ensured that each required data attribute could be extracted accurately across different lecturer profiles.

In addition to identifying relevant elements, the URL patterns and pagination mechanisms were also examined. Each lecturer profile may contain multiple pages of SCOPUS-indexed publications, and these pages follow a predictable structure. By understanding this pattern, the scraping process can systematically access all available pages without manual intervention.

To improve reliability, several lecturer profiles were analyzed to confirm the consistency of the HTML structure. This step was necessary to reduce potential errors caused by structural variations between pages [25]. The results of this analysis served as the foundation for designing an effective scraping strategy.

2.2 Scraping Module Design

The scraping module was developed using PHP with the Simple HTML DOM library. This library was chosen because the SCOPUS publication pages on SINTA are rendered using static HTML structures, meaning that JavaScript execution is not required to access the target data. Therefore, this parser provides a practical and efficient solution for the objectives of this study.

Functions such as `file_get_html()`, `find()`, and `plaintext()` were used to extract specific data from HTML elements. The `file_get_html()` function retrieves the webpage content, while `find()` is used to locate elements based on their tags or class attributes. The extracted data are then processed into readable text using `plaintext()`, enabling structured data collection.

Furthermore, the module was designed to navigate through all available pages via pagination. The extracted data from each page were temporarily stored in arrays, allowing further filtering and processing before final use [26]. For platforms that rely on dynamic rendering, frameworks such as Selenium would be more appropriate alternatives [27].

2.3. Iterative Scraping Process

In a conventional scraping procedure, data extraction is typically performed once per page. On the SINTA platform, each page of the SCOPUS publication section displays a maximum of ten publication entries. The standard scraping process employed in SINTA is illustrated in Figure 1.

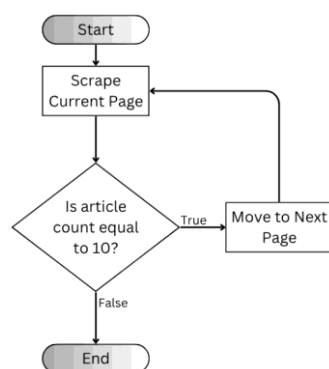


Figure 1. Standard Scraping on SINTA

As shown in Figure 1, the standard method retrieves data sequentially page by page, but the process terminates once a page contains fewer than ten entries. This approach is straightforward and efficient for stable data sources, yet it has clear limitations in the context of SINTA. Specifically, newly added publications may not be captured in a single pass, and inconsistencies in page rendering may lead to incomplete datasets.

These limitations are mainly caused by SINTA organizing SCOPUS publications across multiple pages. Although articles are generally ordered by publication year, the placement of articles within the same year is often inconsistent and appears random. An example of such ordering can be seen in Figure 2, which illustrates two profiles, each containing two pages of publications.

As shown in Figure 2, differences in article distribution across pages have a significant impact on scraping reliability. In Profile 1, articles from the same year (e.g., 2021) are scattered across two different pages, causing the standard scraping method to frequently produce duplicates or miss entries. In contrast, Profile 2 demonstrates a more structured pagination, where the

year 2021 ends on the first page and the following page begins with an earlier year. This clear separation enables data to be captured more consistently in a single scraping pass. This observation explains why the standard method sometimes works for one profile but fails for another. It should be emphasized that these profiles are presented solely for illustrative purposes to highlight ordering irregularities and were not included in the evaluation experiments.



Figure 2. Article Distribution by Publication Year in Two Profiles

To address the risk of missing data or duplicates caused by such random distribution, an iterative scraping approach is proposed. This method performs repeated retrievals across all available pages while simultaneously applying duplicate checks. The overall workflow of this iterative procedure is presented in Figure 3.

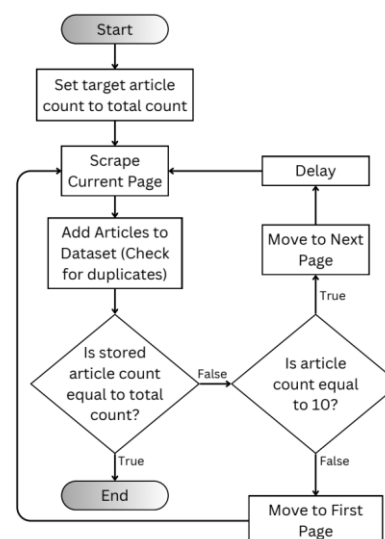


Figure 3. Iterative Scraping on SINTA

Figure 3 illustrates the workflow of the iterative data retrieval process on SINTA. The process begins at the Start stage, followed by setting the target number of articles (set target article count to total count). This value is obtained from the total publication information displayed on the lecturer's profile page and is retrieved only once at the beginning of the process.

Next, the system scrapes the current page, where all article data on the currently accessed page are extracted. The extracted data are then processed in the added articles to dataset (check for duplicates) stage. At this stage, each article is examined to ensure that no duplicate entries are stored. Duplicate checking is performed by comparing the article title with those already stored in the dataset. If the article does not exist in the dataset, it is added; otherwise, the system proceeds to the next article.

After all articles on the page have been processed, the system evaluates the condition is stored article count equal to total count. If the number of stored articles matches the target count, the process terminates (End). Otherwise, the system proceeds to the next decision stage, is article count equal to maximum per page, to determine whether additional pages are likely to exist.

If the number of articles on the current page reaches the maximum capacity per page, the system proceeds to move to next page. Otherwise, the system returns to move to first page to restart the process from the beginning. Before navigating between pages, a delay is introduced to reduce the risk of being detected as a bot or blocked by the server. This delay is implemented using the sleep() function in PHP [28], [29].

This process is performed iteratively. The system repeatedly executes the data extraction cycle across available pages. In each iteration, the retrieved data are compared with the existing dataset to identify any new records. The process continues until no additional articles are found. At this point, the total number of stored records has reached the predefined target.

To demonstrate data instability on the SINTA platform, the scraping experiment was conducted repeatedly on the same lecturer profile. A total of ten independent trials were performed using an identical scraping configuration. This repeated-access approach was designed to observe variations in the retrieved data

caused by inconsistent and seemingly random ordering of articles across pages, particularly when articles from the same publication year are distributed across multiple pages.

2.4 Validation and Evaluation

Upon completion of the scraping process, the collected dataset was validated through manual cross-checking against the actual display of selected lecturer profiles on the SINTA platform. This validation procedure served as a data quality assurance step, ensuring that all entries were complete, accurate, and free from duplication.

The effectiveness of the proposed approach was subsequently evaluated by conducting a comparative analysis between standard (single-pass) scraping and the iterative scraping method. The evaluation employed several parameters, including the total number of unique entries retrieved, the number of duplicate records identified and removed, and the total number of pages successfully extracted. The outcomes of this evaluation provided empirical evidence of the advantages of the iterative scraping approach in producing more accurate, comprehensive, and reliable publication datasets.

3. Results and Discussions

The experiment focused on the SINTA profile with the highest score at the time of the study, specifically targeting publications indexed in SCOPUS. This profile contained a total of 240 publications distributed across 24 pages. Data collection was performed using an iterative scraping method, while recording the results of each iteration to enable comparison with the single-pass (standard) scraping approach.

Prior to scraping, the structure of each data element to be extracted was explicitly defined. Table 1 presents the list of data elements and the extraction techniques employed.

Table 1. Publication Data Element Structure and Extraction Techniques

No	Element	Selector	Description
1	title	\$el = \$pub->find('.ar-title', 0); \$el ? trim(\$el->plaintext) : "	Extracts article title from the .ar-title element if available.
2	author	get_string_between(\$pub, '>Creator : ', '')	Extracts author name from raw HTML.
3	publisher	\$el = \$pub->find('.ar-pub', 0); \$el ? trim(\$el->plaintext) : "	Extracts publisher from .ar-pub element.
4	quartile	\$el = \$pub->find('.ar-quartile', 0); explode(' as ', \$el->plaintext)[0]	Retrieves the journal quartile from before the "as" keyword.
5	type	\$el = \$pub->find('.ar-quartile', 0); explode(' as ', \$el->plaintext)[1]	Retrieves the publication type from after the "as" keyword.
6	year	\$el = \$pub->find('.ar-year', 0); \$el ? trim(\$el->plaintext) : "	Extracts publication year from .ar-year element.
7	citation	\$el = \$pub->find('.ar-cited', 0); \$el ? trim(\$el->plaintext) : "	Extracts citation count from .ar-cited element.

Table 1 presents the target data elements extracted during the scraping process along with the techniques applied. Each element corresponds to a specific publication metadata field, including article title, author name, publisher, quartile, publication type, year, and citation count. Most elements were extracted using standard DOM-based selectors, as they are represented

by dedicated HTML classes (e.g., .ar-title, .ar-pub, .ar-year).

In contrast, the author information on the SINTA platform is presented only as a single author, namely the first author, labeled as Creator. This information is embedded within an anchor (<a>) tag that does not have a specific or consistent class attribute and is preceded by the textual label "Creator :". As a result, standard

DOM selectors cannot reliably isolate the author name alone. Therefore, a custom string extraction function, `get_string_between()`, was employed to retrieve the author name by capturing the text located between the markers “>Creator :” and “” in the raw HTML.

Initially, the same string-based extraction approach was also applied to retrieve the publication type using the keyword “as”. However, this approach proved unreliable because the term “as” may also appear in other HTML contexts, such as article titles. To address this limitation, publication quartile and type were

instead extracted directly from the .ar-quartile element, which follows a consistent textual structure (e.g., “Q3 as Journal”). The string was then split at the keyword “as”, yielding two distinct values: the quartile (e.g., Q3) and the publication type (e.g., Journal). This DOM-based extraction strategy provides more consistent and accurate results than free-text parsing.

Following the definition of all data elements, the scraping experiments were carried out and the results systematically recorded. Table 2 reports the number of articles retrieved in each iteration across ten trials.

Table 2. Article Count per Iteration across 10 Trials

Iteration	Trial									
	1	2	3	4	5	6	7	8	9	10
1	166	165	176	153	172	185	179	170	154	177
2	212	216	221	213	219	212	216	213	207	214
3	220	229	232	219	234	229	231	224	234	219
4	232	229	240	236	239	236	236	228	235	233
5	239	235	—	237	239	239	239	237	238	236
6	239	236	—	239	240	240	240	239	239	238
7	240	236	—	240	—	—	240	239	239	240
8	—	236	—	—	—	—	—	239	239	—
9	—	239	—	—	—	—	—	240	240	—
10	—	239	—	—	—	—	—	—	239	—
11	—	240	—	—	—	—	—	—	240	—

Table 2 shows that the number of iterations required to retrieve all publications varied, ranging from four to eleven. Although theoretically possible to collect all data in a single pass, the probability of success is low. These results confirm that iterative scraping is more effective in ensuring complete data retrieval without missing or duplicate entries.

Furthermore, Table 2 highlights substantial variation in the number of articles obtained during the first iteration. On average, only 169.7 out of 240 articles (70.7%) were captured in the initial trial. The results of the first iteration are visualized in Figure 4.

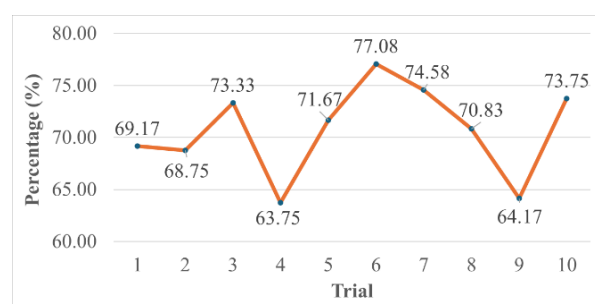


Figure 4. Percentage of Publications Collected in the First Iteration

Figure 4 illustrates that relying on a single scraping pass can result in up to 30% of data being lost or duplicated. Compared to the standard single-pass scraping, which retrieved only 70.7% on average, the iterative approach consistently reached 100% across all trials. A closer examination of Table 2 reveals that, in Trial 9, one article was only retrieved on the fifth iteration. This article belonged to a cluster of 104 publications from 2024, distributed across 12 pages (pages 2–13). This distribution underscores the importance of iterative scraping in ensuring comprehensive and accurate data

collection, particularly when publications from the same year are scattered across multiple pages.

The variation in the number of required iterations, ranging from as few as four to as many as eleven, can be attributed to the dynamic ordering of articles within the same publication year and the frequent reshuffling of entries across pages. Profiles with a higher concentration of articles from the same year such as the 104 publications from 2024 observed in Trial 9 tend to require more iterations, as the scraper must repeatedly adjust to capture records that shift between pages.

Conversely, profiles with a more balanced distribution of publication years stabilize more quickly, reducing the number of iterations required. These results suggest that the iterative method is sensitive to the density and placement of year-based clusters within the pagination structure. From a practical standpoint, researchers should anticipate multiple iterations when working with large datasets in which articles are unevenly distributed, and they should implement automated stopping criteria (e.g., termination once no new articles are detected) to minimize computational overhead while ensuring dataset completeness.

From a practical perspective, the iterative scraping method offers clear advantages for real-world applications that require reliable and comprehensive data collection from SINTA or similar academic indexing platforms. Its primary benefit lies in its ability to minimize missing or duplicate records when dealing with large, complex profiles where articles from the same year are scattered across multiple pages. This ensures higher data integrity, which is essential for downstream tasks such as bibliometric analysis,

institutional performance evaluation, and research trend monitoring.

However, the method also presents certain disadvantages. Compared to standard scraping, it requires more computational resources and longer execution times due to the repeated iterations. Additionally, the iterative approach may generate temporary redundant requests to the server, which could raise ethical and technical considerations if not managed with appropriate throttling mechanisms. Therefore, while the iterative method significantly improves data completeness, its deployment must balance accuracy with efficiency. In practice, this balance was achieved by implementing automated stopping criteria and integrating request delays using the sleep() function. Such throttling mechanisms are essential not only to optimize performance but also to adhere to ethical standards of web scraping, thereby minimizing the risk of server overload or violation of platform policies.

4. Conclusions

This study demonstrates that the structural characteristics of publication data within SINTA lecturer profiles, particularly those indexed in SCOPUS, can result in overlaps when articles from the same publication year are distributed across multiple pages. Such conditions reduce the reliability of the standard single-pass scraping method, which on average captured only 70.7% of the total publications in the first iteration and frequently produced missing or duplicate records. In contrast, the iterative scraping method consistently achieved complete data retrieval across all ten trials, successfully collecting 100% of publications. The number of iterations required, however, varied between four and eleven, reflecting the influence of page dynamics and publication distribution. A notable case was observed in Trial 9, where a single article within a cluster of 104 publications from 2024 was retrieved only on the fifth iteration due to its dispersion across 12 pages. These findings underscore the superior adaptability of the iterative approach in handling dynamic pagination structures and uneven year-based clustering. Overall, the iterative scraping method offers a more robust and dependable strategy for ensuring comprehensive and accurate data extraction from platforms such as SINTA. Although it entails greater computational demands compared to standard scraping, its ability to simultaneously guarantee data completeness and accuracy makes it particularly suitable for high-stakes applications, such as institutional performance evaluation, bibliometric assessment, and research policy decision-making, where incomplete or erroneous data may lead to misleading conclusions and unfair evaluations.

Acknowledgements

The author declares no conflict of interest. This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

References

- [1] N. Adila, "Implementation of Web Scraping for Journal Data Collection on the SINTA Website," *Sinkron*, vol. 7, no. 4, pp. 2478–2485, Oct. 2022, doi: [10.33395/sinkron.v7i4.11576](https://doi.org/10.33395/sinkron.v7i4.11576).
- [2] A. Ulfah and I. Najiah, "Implementasi Web Scraping pada Situs Jurnal Sinta Menggunakan Framework Selenium Webdriver Python," *JIKA (Jurnal Inform.)*, vol. 7, no. 1, p. 29, Feb. 2023, doi: [10.31000/jika.v7i1.7037](https://doi.org/10.31000/jika.v7i1.7037).
- [3] E. D. Widiyanto, Hadiyanto, T. Mantoro, and R. S. Wardoyo, "The promotion of university journals published by Universitas Diponegoro, Indonesia, from 2018 to 2024: a descriptive study," *Sci. Ed.*, vol. 12, no. 1, pp. 43–49, Feb. 2025, doi: [10.6087/kcse.353](https://doi.org/10.6087/kcse.353).
- [4] M. K. Hidayat, D. Sugianto, R. Fitriana, and Y.-C. Liang, "Business intelligence system model to measure the performance of lecturers' scientific publications," *TELKOMNIKA (Telecommunication Comput. Electron. Control.)*, vol. 23, no. 4, p. 954, Aug. 2025, doi: [10.12928/telkomnika.v23i4.26221](https://doi.org/10.12928/telkomnika.v23i4.26221).
- [5] E. D. Widiyanto, Hadiyanto, T. Mantoro, and A. S. Nugroho, "Subject area mapping of Indonesian journals from GARUDA (Garba Rujukan Digital) and SINTA (Science Technology Index) databases: a descriptive study," *Sci. Ed.*, vol. 12, no. 2, pp. 175–182, Aug. 2025, doi: [10.6087/kcse.382](https://doi.org/10.6087/kcse.382).
- [6] A. Nugrohadhi, "Pemanfaatan Publikasi Ilmiah di Perguruan Tinggi," *Pustakaloka*, vol. 9, no. 2, Nov. 2017, doi: [10.21154/pustakaloka.v9i2.1087](https://doi.org/10.21154/pustakaloka.v9i2.1087).
- [7] A. Mubarak and P. Istiana, "Mengkaji Publikasi Dosen Menggunakan Analisis Bibliometrik," *Media Inf.*, vol. 31, no. 2, pp. 146–156, Nov. 2022, doi: [10.22146/mi.v31i2.5402](https://doi.org/10.22146/mi.v31i2.5402).
- [8] A. Nurohman and A. Aprillilah, "Riset kebijakan publikasi repositori institusi Perguruan Tinggi Keagamaan Islam Negeri Indonesia," *AL-KUTTAB J. Kaji. Perpustakaan, Inf. dan Kearsipan*, vol. 6, no. 1, pp. 52–70, 2024, doi: [10.24952/ktb.v6i1.10443](https://doi.org/10.24952/ktb.v6i1.10443).
- [9] Y. S and D. F. Prakash, "Exploring Web Scraping Techniques for Data Extraction in E-Commerce Platforms: A Comprehensive Analysis," *Int. Res. J. Mod. Eng. Technol. Sci.*, vol. 6, no. 3, Mar. 2024, doi: [10.56726/IRJMET50725](https://doi.org/10.56726/IRJMET50725).
- [10] J. Ward, *Instant PHP Web Scraping*. Packt Publishing, 2013.
- [11] U. Kumaresan and K. Ramanujam, "A Framework for Automated Scraping of Structured Data Records From the Deep Web Using Semantic Labeling," *Int. J. Inf. Retr. Res.*, vol. 12, no. 1, pp. 1–18, Nov. 2021, doi: [10.4018/IJIRR.290830](https://doi.org/10.4018/IJIRR.290830).
- [12] M. Dogucu and M. Çetinkaya-Rundel, "Web Scraping in the Statistics and Data Science Curriculum: Challenges and Opportunities," *J. Stat. Data Sci. Educ.*, vol. 29, no. sup1, pp. S112–S122, Jan. 2021, doi: [10.1080/10691898.2020.1787116](https://doi.org/10.1080/10691898.2020.1787116).
- [13] A. Rahmatulloh and R. Gunawan, "Web Scraping with HTML DOM Method for Data Collection of Scientific Articles from Google Scholar," *Indones. J. Inf. Syst.*, vol. 2, no. 2, pp. 95–104, Feb. 2020, doi: [10.24002/ijis.v2i2.3029](https://doi.org/10.24002/ijis.v2i2.3029).
- [14] A. F. Putri, G. Manik, F. Nabila, and N. Chamidah, "Implementasi Scraping Google Scholar Menggunakan HTML DOM Untuk Pengumpulan Data Artikel Dosen UPN Veteran Jakarta Berbasis Web," in *Prosiding Seminar Nasional Mahasiswa Bidang Ilmu Komputer dan Aplikasinya*, 2021, vol. 2, no. 1, pp. 668–678.
- [15] M. R. Rafsanjani, "ScrapPaper: A web scraping method to extract journal information from PubMed and Google Scholar search result using Python," *bioRxiv*, Mar. 2022, doi: [10.1101/2022.03.08.483427](https://doi.org/10.1101/2022.03.08.483427).
- [16] M. Koprari and W. S. Putra, "Implementasi Web Scraping pada Google Cendekia sebagai Sarana Profiling Penelitian Dosen," *Sci. Tech J. Ilmu Pengetah. dan Teknol.*, vol. 9, no. 1, pp. 59–72, Feb. 2023, doi: [10.30738/st.vol9.no1.a14188](https://doi.org/10.30738/st.vol9.no1.a14188).
- [17] Y. A. Mahmood and B. Mahmood, "A Web Scraper for Data

- Mining Purposes,” *SISTEMASI*, vol. 13, no. 3, p. 1243, May 2024, doi: [10.32520/stmsi.v13i3.4107](https://doi.org/10.32520/stmsi.v13i3.4107).
- [18] Y. I. Sulistya, A. C. Wardhana, M. Istighosah, and A. Riyandi, “Automated Google Scholar Crawling with a Web-Based Tool for Publication Data Management,” *J. Teknol. Dan Sist. Inf. Bisnis*, vol. 6, no. 4, pp. 768–773, Oct. 2024, doi: [10.47233/jteksis.v6i4.1604](https://doi.org/10.47233/jteksis.v6i4.1604).
- [19] S. R. Shrestha, L. Olivetti, S. Pandey, K. Worou, and E. Raffetti, “The Citation Gap: An overview of academic output in the field of Natural Hazards and Climate Extremes analyzed through Google Scholar data.” Mar. 2025. doi: [10.31235/osf.io/az784_v1](https://doi.org/10.31235/osf.io/az784_v1).
- [20] J. Räsänen, “Missing references and citations at Google Scholar,” *Bioethics*, vol. 39, no. 4, pp. 399–400, May 2025, doi: [10.1111/bioe.13403](https://doi.org/10.1111/bioe.13403).
- [21] F. Hanifah, A. Rahmatulloh, and R. N. Shofa, “Implementasi Web Scraping Dalam Pengembangan Aplikasi Super Untuk Menunjang Kegiatan Tugas Akhir,” *J. Nas. Teknol. dan Sist. Inf.*, vol. 11, no. 3, pp. 369–376, Jan. 2026, doi: [10.25077/TEKNOSI.v11i3.2025.369-376](https://doi.org/10.25077/TEKNOSI.v11i3.2025.369-376).
- [22] Y. Sahria, “Implementasi Teknik Web Scraping pada Jurnal SINTA Untuk Analisis Topik Penelitian Kesehatan Indonesia,” in *URECOL (University Research Colloquium)*, 2020, pp. 297–306.
- [23] L. M. Purnomo and M. Ayub, “Analisis Data Hasil Web Scraping untuk Menentukan Kualitas Jurnal Ilmiah,” *J. Strateg.*, vol. 3, no. 1, pp. 122–132, 2021.
- [24] S. I. Nurhafida, “Optimasi Query Database Webscraping pada Jurnal Sinta,” *JATISI (Jurnal Tek. Inform. dan Sist. Informasi)*, vol. 9, no. 3, pp. 2521–2532, Sep. 2022, doi: [10.35957/jatisi.v9i3.2806](https://doi.org/10.35957/jatisi.v9i3.2806).
- [25] T. M. Fahrudin et al., “An Improved Reference Paper Collection System Using Web Scraping with Three Enhancements,” *Futur. Internet*, vol. 17, no. 5, p. 195, Apr. 2025, doi: [10.3390/fi17050195](https://doi.org/10.3390/fi17050195).
- [26] M. I. Al Amin, “PARSING DOM Method Implementation for Web Scraping to Identify Pornographic Websites,” *JESII J. Elektron. Sist. Inf.*, vol. 2, no. 1, pp. 171–178, Jul. 2024, doi: [10.31848/jesii.v2i1.3414](https://doi.org/10.31848/jesii.v2i1.3414).
- [27] B.-V. Truong, L. T. T. Nguyen, P. Pham, and B. Vo, “HTMLDownloader: An open-source tool for dynamic web scraping and archiving using WebView2,” *SoftwareX*, vol. 32, p. 102373, Dec. 2025, doi: [10.1016/j.softx.2025.102373](https://doi.org/10.1016/j.softx.2025.102373).
- [28] S. jahro Maulidiyah and A. I. Syahyadi, “Analysis Of E-Commerce Product With Web Scraping Technique,” *CoreID J.*, vol. 3, no. 1, pp. 37–46, Mar. 2025, doi: [10.60005/coreid.v3i1.90](https://doi.org/10.60005/coreid.v3i1.90).
- [29] R. Ulloa, F. Mangold, F. Schmidt, J. Gilsbach, and S. Stier, “Beyond time delays: how web scraping distorts measures of online news consumption,” *Commun. Methods Meas.*, pp. 1–22, Mar. 2025, doi: [10.1080/19312458.2025.2482538](https://doi.org/10.1080/19312458.2025.2482538).