A Citation Data Collector Tool of Author's Profiles in Scopus Based on Web and Application Programming Interface (API)

by Adian Fatchur Rochim

A Citation Data Collector Tool of Author's Profiles in Scopus Based on Web and Application Programming Interface (API)

A Citation Data Collector Tool of Author's Profiles in Scopus Based on Web and Application Programming Interface (API)

Adian Fatchur Rochim[1], Tendi Nugeraha Wijaya[2], Dania Eridani[3]

Department of Computer Engineering, Diponegoro University, Tembalang, Semarang, Indonesia, 50275

E-mail: [1]adian@ce.undip.ac.id, [2]tendinw@students.undip.ac.id, [3]dania@ce.undip.ac.id

Abstract. Improving the quality of publications will enhance teaching, support community service, and improve the campus reputation. One of the measurements of research quality can be observed by looking at the number of citations. The more research cited, the greater the effect it will have on other research. In addition, the publication of research results is used by most researchers in relation to the promotion of their academic positions. The position of the author in the research affects the weight value of his/her publication. At this time, no application can be used to classify the author's position of each publication produced. This paper aims to create an application that acted as a search engine to retrieve and collect researcher data along with a list of their publications from the Scopus database. The application uses a REST API or Web Service of Scopus. Web Service allows the integration of various systems with different platforms or architectures for sharing data and services. This application was built using the PHP and JavaScript programming languages. The application's development uses the Agile Software Development method. Users can also save the search results for the researcher data and researcher documents list in the MySQL database. Furthermore, this application also allows users to download the search result in their computer devices. The System Usability Scale (SUS) method was used to test the application's usability and gained 80 as the final score.

1. Introduction

Measuring the quality of research activities is essential to determine an impact on the development of science, technology, social, and economic[1]. Good quality research can also improve the development of teaching, support community service, and enhance the campus's reputation. In today's digital era, an alternative to measure a scholarly output of research can be observed by looking at a number of citations and papers of authors' papers. The more a paper cited, the influence of the research on science and technology development is even more significant[2]. Similarly, the publication of research results is used by most researchers concerning the promotion of academic positions. The most significant weight of publication results is when, in publication, the researcher is the main author.

To obtain research publication data, researchers can open the citation database or scientific journals, such as Google Scholar, Scopus, Crossref, Web of Science, and many more. In the previous research, researchers also used the web scraping and Application Programming Interface (API) method to obtain data[2][3][4]. The use of web scraping or API is quite complicated to analyze the research. Another study used as a reference in this paper is the creation of a search engine to search researcher profiles and research publications[5]. It used a web scraping method on other literature databases to collect data.

Based on previous studies that have been mentioned above, there has not been found an application that can be used to classify the author number sequence of each published publication. And to make it easier for researchers to analyze their citation data statistics, an application that is integrated with the API is needed. The application can search for research statistics from database sources. Researchers can do a real-time live search for statistics of a study. This application's need is important for researchers to be used in finding and classifying the weight value of each publication produced. This application is important so that the academic promotion review team can easily classify each author's publication output.

In this paper, a search application is made to retrieve and collect research data from citation databases or scientific journals. This system uses REST API (Representational State Transfer Application Programming Interface) or Web Service. In this application, the database used is Scopus. The application connects to the Scopus database using the API provided by Elsevier. This application also allows users to save search results of researchers and researcher documents into a database. The database used in this application is MySQL. This application was built using the PHP and JavaScript programming languages.

This paper consists of six sections, i.e., introduction, literature review, methodology, result and discussion, acknowledgments, and conclusions. The introduction section has been presented above. The next section of this paper is the literature review regarding the concepts and theories used during this research. Then there will be sections of the methodology used in designing the application, results, and discussion about the application, acknowledgments, and the conclusions of this paper.

2. Literature Review

2.1. Application Programming Interface (API)

Representational State Transfer (REST) is a software architectural style for distribution or communication of web services focusing on system resources, first presented by Roy Fielding in 2000 in his dissertation[6]. Meanwhile, Application Programming Interface (API) is code for connecting one application with data and services from another application by giving all necessary permissions and allowing two software programs to communicate with each other[7]. REST API is a web service that uses Uniform Resource Identifier (URI), Hypertext Transfer Protocol (HTTP), and JavaScript Object Notation (JSON) for data format[7].

2.2. Scopus and Scopus API

Scopus is a database containing abstracts and citations of academic journals owned by Elsevier. Scopus contains approximately 24,600 titles from 5,000 publishers[8]. Scopus uses sophisticated tools and analytics to produce precise citation results, detailed researcher profiles, and insights that drive better decisions, actions, and outcomes. Scopus generates precise citation search results and automatically updates researchers and institutions' profiles, creating wider connections between people, published ideas, and organizations.

Scopus has an API that allows users to retrieve publication data via HTTP requests[9]. The Scopus API works in a machine-readable format that enables the software to find articles, authors, and institutions in the Scopus database[9]. This paper uses two Scopus API models. The first model is the author's search, which is used to search for researchers' profiles on Scopus. The second model is Scopus Search, which is used to search for researcher documents on Scopus.

2.3. Previous Research

In 2016, Adian conducted a research that aims to show a researcher's performance from their published articles. The application produced can crawl the data from Google Scholar, Crossref, and Mendeley[2]. In this application, the method used to retrieve data from Google Scholar is scrapping. In contrast, the method used to retrieve data from Crossref and Mendeley is by using API.

In 2017, Adian conducted a research that aims to analyze the number of readers of articles published by 100 Indonesian researchers in Mendeley[3]. This research used Mendeley's API to calculate the number of readers for each article. Then it combines with Google Scholar citations using the scraping method.

In 2017, Aditya conducted a research that aims to design a search feature with search engine for staff website of University of Indonesia. The application can perform a search for profile and publications of researchers[5]. This application collecting the data by web scraping on other literature databases.

3. Methodology

The Agile Software Development method is used to design the Data Collection Application. Its characteristics are to focus the developers on the developers' working conditions, the working software, the customers, and addressing changing requirements instead of focusing on detailed systems development processes, tools, all-inclusive documentation, legal contracts, and detailed plans[10]. Compared to other methods, the agile method was chosen because it is suitable for developing this application. In developing this application, the developer must be responsive in accepting changing requirements, the user is actively involved in the development, and focuses on working software instead of documentation.

After the application development is completed, the application was tested using two methods, namely the black box and the System Usability Scale (SUS) testing. Black box testing is used to test the functionality of the application, while SUS is used to test the usability of the application. SUS testing is done using ten respondents. Respondents were grouped based on gender, age, the latest education, and current occupation.

Based on the development method used, this section consists of four subsections, namely, planning, analysis, design, and implementation[10]. The testing result of the application will be discussed in the Result and Discussion section.

3.1. Planning

Software planning begins with determining the scope of the software, then continues by estimating the required resources. This app requires Scopus resources as a data provider. The application is built on a web basis using the PHP 7.2.10 and MySQL 5.0.12 to store and maintain the data. The local servers used to run the database are Apache and MySQL using XAMPP. Sublime Text 3 is used as a text editor. To access the application, it uses Google Chrome as a web browser. To run the search, it needs a university network, so to be used outside the university, it used OpenVPN application. This application is built by a writer who has the role of a full stack developer. It takes three months to build the Data Collection Application.

3.2. Analysis

After studying the references, there has not been found an application that can be used to classify the author number sequence of each published publication. Based on the analysis that has been done, we need an application that is integrated with a citation and scientific journal database that can search for researcher data and research documents. The application needed can also determine the author's position in the research. In this paper, a search application will be made to search for researchers' data and their publication from the Scopus database. This application uses the REST API to integrate the application with the Scopus database. Users can search for researcher data and a list of researcher documents on Scopus.

3.3. Design

The architecture of the Data Collection Tool can be seen in Figure 1. The architecture divides the system into four main components. The first part is the application that provides an interface for the user to send the search parameters, receive the information, and data processing. The second component is a web

service that contains REST API which is used to retrieve data from the resource. The third component is the resource as a data provider. The last component is the database used to store the data.
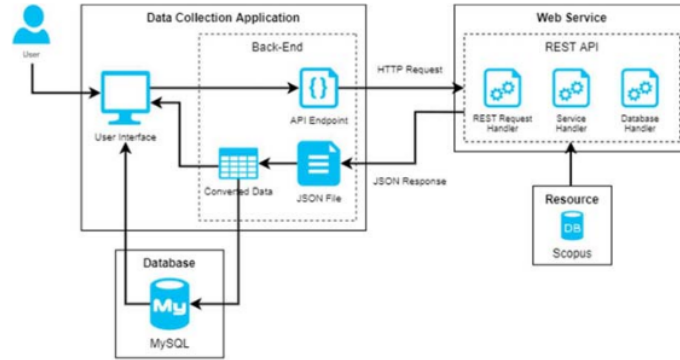
Figure 1. The Architecture of Data Collection Application

The main feature of the application is as a search engine. The search process starts when the user enters the keywords in a search form. The form will send parameters by adding data to the URL. API is used to retrieve data from a resource. This application uses Scopus API to access Scopus data in real-time.

To interact with the Scopus API, the system uses the cURL library. cURL is used to make an HTTP request. HTTP GET request method is used to get data information. The final result of the use of Scopus API and the cURL library is JSON data. The JSON data will be converted into an array. Furthermore, the array will be displayed in a table on the application interface so that it is easy for users to understand. Not only can be displayed, the search result that has been converted into an array can be saved into MySQL database. The process of inputting values from the array into the database uses SQL commands and PHP scripts.

Not all the data obtained from the Scopus API will be displayed in the application. This application will display data commonly used to analyze research, such as the number of citations, the number of authors, and the author's position. The data from the Scopus API that will displayed can be seen in Table 1 and Table 2.

Table 1. Author Search Used Field

No	Field	Name	Description
1.	preffered-name, surname	Name	First and last name of author
2.	affiliation-current	Affiliation	The author organization
3.	document-count	Documents	Number of author documents

Table 2. Scopus Search Used Field

No	Field	Name	Description
1	dc:title	Title	Article title
2	prism:doi	DOI	Digital Object Identifier
3	citedby-count	Cited by	Number of articles cited
4	author-count, @total	Number of Author	Number of article authors

The fields mentioned above can be obtained directly from the search result of the Author Search API and Scopus Search API. This application will display the researcher's position in research, but there is no author position field in the search result of Scopus Search API. Then a PHP script is used to determine

the position of the author. The flowchart to determine the position of the author is shown in Figure 2. The author's name is searched in array using array_search function of PHP. It is searched on an array result from the JSON conversion from the Scopus Search API results The array_search result is the index number that contains the author's name. Indexes in arrays always start at 0. Then, to determine the author's position, the formula index+1 is used. For example, author x is the first author of the research. In the array, the author is at index 0. Thus, to display the position 1, the formula 0+1=1 is used.
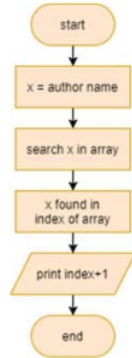
Figure 2. Flowchart to determine the position of the author

In the Scopus Search API, the maximum number of authors that can be displayed is 100. So that if the research has more than 100 authors, the author's position in the research can't be determined. In this application, when the author's position cannot be determined, it will display "N/A".
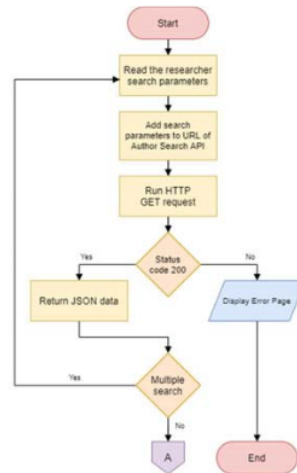
3.4. Implementation
The implementation is the last stage in the software development cycle. This stage consists of three parts, namely API Implementation, database implementation, and application implementation. The first two-part will be described in the next subsubsection, while the application implementation will be discussed entirely in the Result and Discussion subsection.

3.4.1. API Implementation. Implementation of the Scopus API is done using PHP. Scopus API is implemented along with cURL, a PHP library that can be used to make HTTP requests. There are two models of Scopus API used, namely Author Search API and Scopus Search API.

Author Search API is used to obtain the profile of the researcher. The parameters used in the Author Search are the first name, last name, and affiliation of the researcher. The flowchart for the implementation of Author Search API can be seen in Figure 3. The search parameters are read, then those will be added to the Author Search API URL. The system runs the HTTP GET request to read or get data from the source. It will get an HTTP status as a response to client requests. If the status code is 200, the server has successfully processed the request. A successful request will return JSON data. If the user performs a search more than once, the system will restart from reading the search parameters.

The second model that is implemented is the Scopus Search API. It uses the Author ID as the search parameter that obtained from the previous use of Author Search API. The flow diagram for the implementation of the Scopus Search API can be seen in Figure 4. The document search parameters will be read and added to the Scopus Search API URL. A HTTP GET request is executed and return a status code. The server successfully processed the request when the status code is 200. The application will check the number of documents. If those are less than 25, the search results will be displayed immediately. However, when the number of documents more than 25, a new HTTP GET request will

be made for the next page. This is because the Scopus Search API will only display the most recent 25 documents in one execution. To display the 26th document onwards, a new HTTP GET request is needed for the 2nd page. This is an iterative process. A successful request will return JSON data.

**Figure 3.** Flowchart for the implementation of Author Search API
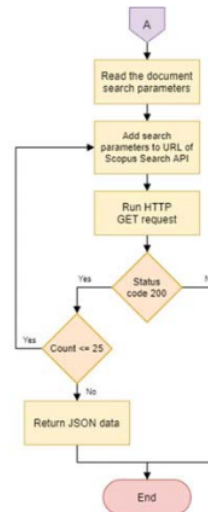
Figure 4. Flow diagram for the implementation of the Scopus Search API

3.4.2. Database Implementation. The database was created using SQL (Structured Query Language) using the MySQL DBMS (Database Management System). The local servers used to run the database are Apache and MySQL using XAMPP. The Data Collection Application database consists of three

tables; there are users table, scopus_author table, and scopus_document table. The tables and their relation in the Data Collection Application database can be shown in Figure 5.
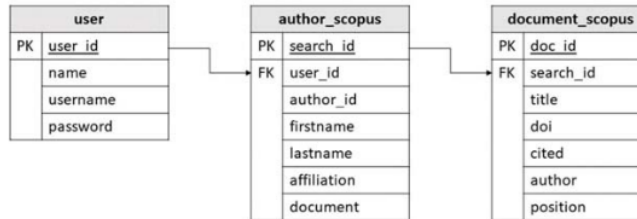
Figure 5. Tables and relation in the Data Collection Application database

4. Result and Discussion

This section will explain the application implementation. The application implementation will display the interface of the application. This section will also explain the test that aims to check whether the application made is following the design objectives that have been previously determined.

4.1. Application Implementation

To run the application, it requires an internet connection of an institution that subscribes to Scopus. The application is accessed via a web browser by entering the address of the application in the address bar. To be able to use the full features of the application, sign in is required for user. User searches for profiles and documents of researchers on the Search Page as shown in Figure 6.

Figure 6. Search Page of Data Collection Application. Users input name, last name, and affiliation of the researcher as search parameters. Author search can be done in multiple at one time.

The search results will be displayed on the Result Page. On that page, there are three options for each researcher, namely the Save Result button, the Download Docs button, and the View Docs button. The Save Result button is used to save the search result into the database. The Download Docs button is used to download the search result into the user's computer device in excel (.xls) format. The View Docs button is used to view the list of researcher's documents. A list of researchers' documents is shown in Figure 7.

Figure 7. A list of researchers' documents. The information displayed is the research title, Digital Object Identifier (DOI), number of citations, number of authors, and author position.

4.2. Application Testing

Black box testing treats class as a black box[10]. In this test, the code's design and structure are unknown by the testers, technicians, and users who will perform this test. Based on the black box testing results that have been done, this application functionally can produce the expected output.

System Usability Scale (SUS) is also used to test the application. SUS is part of usability testing, which focuses on testing that is done directly by the end-user. SUS is a simple, ten-item scale giving a global view of subjective assessments of usability[11]. Respondents were asked to rate the usefulness of the product on a 5-point scale numbered from 1 (strongly disagree) to 5 (strongly agree)[12]. SUS testing is done in this application using ten respondents as the test sample. The processed SUS data can be seen in Table 3.

Table 3. The Processed SUS Data. The Data Collection Application has an average score of 80. Based on this assessment results, this application can be used easily by end-users as a software that can help users find and analyze the statistics of a researcher.

No	Respondent	Calculated Score										Total	Score
		Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10		
1	Respondent 1	3	4	4	4	4	4	4	4	3	1	35	88
2	Respondent 2	3	3	3	3	4	4	3	3	4	3	33	83
3	Respondent 3	3	4	4	3	4	4	4	4	3	0	33	83
4	Respondent 4	4	4	4	4	4	4	4	4	4	4	40	100
5	Respondent 5	3	3	4	4	4	2	2	4	4	3	33	83
6	Respondent 6	4	4	4	4	4	3	4	4	4	4	39	98
7	Respondent 7	3	3	3	3	3	3	3	3	2	3	29	73
8	Respondent 8	3	1	2	2	3	1	2	1	2	1	18	45
9	Respondent 9	3	3	3	3	3	3	3	3	3	1	28	70
10	Respondent 10	4	3	4	3	3	3	2	3	4	1	30	75
	Average Score (Final Result)												80

5. Conclusions

Based on the results of the analysis, design, implementation, and testing that have been done, it can be concluded that this study has succeeded in building the Data Collection Application. The application

built has been able to fulfil its primary function to search for researcher data. The search for researcher data is a search for profiles and research documents on Scopus. This study succeeded in integrating an application with the Scopus API. The API is used as data sources in the researcher's data search. This application can be used to download the search results of researchers' documents into the user's computer device. In addition, this application is also successful in saving search results for research documents on Scopus in the MySQL database. Based on the SUS testing that has been done, this application has a final score of 80 means this application is easy to use by end-users in finding the statistics of a researcher.

References
[1] Rochim A F, Muis A and Sari R F 2018 Improving fairness of H-index: RA-index *DESIDOC J. of Library & Information Technol.* **38** pp 378–86
[2] Rochim A F and Sari R F 2016 An exploration of Mendeley reader and Google Scholar citation for analysing indexed article *Asea Uninet Sci. Plenary Meet* pp 78–85
[3] Rochim A F and Sari R F 2017 Evaluation of articles published in Mendeley and CrossRef in relation to the Google Scholar pages *ARPN J. Eng. Appl. Sci.* **12**
[4] Alberto M M, Enrique O M, Mike T and Emilio D L C 2018 Google Scholar, Web of Science, and Scopus: A systematic comparison of citations in 252 subject categories *J. Infometrics* **12** pp 1160–77
[5] Putra A 2017 *Pengembangan Prototipe Fitur Pengumpulan dan Pencarian Data Profil dan Publikasi Penelitian pada Website Staff Universitas Indonesia* Undergraduate thesis (Depok: University of Indonesia)
[6] Fielding R 2000 *Architectural Styles and the Design of Network-based Software Architecture* PhD dissertation (California: University of California)
[7] Benharosh J 2018 *What is REST API? in plain English* Retrieved on 20 July 2020 from https://phpenthusiast.com/blog/what-is-rest-api
[8] Elsevier 2020 *About Scopus - Abstract and Citation Database* Retrieved on 20 July 2020 from https://www.elsevier.com/solutions/scopus
[9] Wan K 2020 *What are Scopus APIs and how are these used?* Retrieved on 24 July 2020 from https://www.elsevier.com/__data/assets/pdf_file/0007/917179/Scopus-User-Community-Germany-API-final.pdf
[10] Dennis A, Wixom B H and Tegarden D 2015 *System Analysis & Design An Object-Oriented Approach with UML* (Danvers: John Wileys & Sons, Inc.)
[11] Brooke J 1996 SUS-a quick and dirty usability scale *Usability Evaluation in Industry* (London: Taylor & Francis) chapter 21 pp 189–94
[12] Sharfina Z and Santoso H B 2016 An Indonesian adaptation of the system usability scale (SUS) *Int. Conf. Adv. Comput. Sci. Inf. Syst.* pp 145–8

A Citation Data Collector Tool of Author's Profiles in Scopus Based on Web and Application Programming Interface (API)

with Identical H-index Values", Journal of Data and Information Science, 2020
Publication

5   Dinan Yulianto, Ahmad Raditya Cahya Baswara, Lugas Alhawariy, Marina Indah Prasasti, Gema Antika Hariadi. "Development of Information and Management System of Student Competition Groups through User-Centered Design Approach", Khazanah Informatika : Jurnal Ilmu Komputer dan Informatika, 2023
Publication

<1 %

6   Jun-Ki Min, , and Sung-Bae Cho. "Mobile Human Network Management and Recommendation by Probabilistic Social Mining", IEEE Transactions on Systems Man and Cybernetics Part B (Cybernetics), 2011.
Publication

<1 %

Exclude quotes	Off	Exclude matches	Off
Exclude bibliography	On		

A Citation Data Collector Tool of Author's Profiles in Scopus Based on Web and Application Programming Interface (API)

FINAL GRADE

/0

GENERAL COMMENTS

Instructor

PAGE 1

PAGE 2

PAGE 3

PAGE 4

PAGE 5

PAGE 6

PAGE 7

PAGE 8

PAGE 9

PAGE 10