DESIGNING A TELEGRAM BOT WITH WEB SCRAPING CAPABILITIES FOR AUTOMATING SCIENTIFIC ARTICLE DOWNLOADS

A. Abdillah[1]., A. Rahmatulloh[1]., N. Widiyasono[1], and R. Rizal[1]

[1]Department of Informatics, Faculty of Engineering, Siliwangi University, Kota Tasikmalaya 46115, Indonesia.

Corresponding Author's Email: [1]randirizal@unsil.ac.id

Article History: Received 12 March 2024; Revised 18 April 2024; Accepted 20 May 2024

ABSTRACT: Smartphones have become indispensable tools for a wide range of daily activities, including those related to academics. They serve not only as devices for communication and social interaction but also as powerful tools for self-study and information gathering. One key resource for academics is scientific journals, which provide valuable insights and information. In recent years, advancements in technology have made it possible to integrate instant messaging and access to scientific articles into a single platform. Telegram, a popular instant messaging service, offers an Open API and Protocol that enables developers to create custom bots with diverse functionalities. One such innovative application is a Telegram bot programmed in Python, designed to facilitate the downloading of research papers and articles from the Sci-Hub website using web scraping techniques. This bot allows users to streamline the process of retrieving multiple articles by submitting a list of URLs or DOIs in a simple .txt file format. The bot demonstrates impressive efficiency, with an average response time of 7.48 seconds to download PDF files. Additionally, its message-response time is highly optimized, averaging just 1.06 seconds. By integrating these capabilities, the bot effectively combines academic research needs with the convenience of instant messaging, providing a practical solution for researchers and students alike. This innovation not only simplifies access to scientific resources but also enhances productivity by minimizing the time and effort required for retrieving academic materials, ultimately supporting scholarly goals.

KEYWORDS: *Journal; Python; Scientific Article; Telegram Bot; Web Scraping*

1.0 INTRODUCTION

The rapid advancement of communication technology has revolutionized how humans interact and access information, eliminating the constraints of distance, space, and time [1], [2]. One significant product of this evolution is the smartphone, a multifunctional device that facilitates various activities, including social interaction, self-directed learning, and academic success [3], [4]. Among the many applications that enhance productivity, Telegram stands out as a popular cloud-based communication platform. Telegram not only supports text messaging but also allows users to send media, documents, and utilize open-license bot APIs [5]. In 2023, Telegram ranked as one of the most downloaded applications in Indonesia, making it a relevant tool for exploration and further development.

In the development of Telegram bots, Python has emerged as a preferred programming language due to its comprehensive libraries, readability, and flexibility [6], [7]. Libraries such as BeautifulSoup and requests enable web scraping, allowing Python to extract and utilize data from websites effectively [8]. The web scraping method, which serves as the primary technique in this research, is used to extract website data automatically, leveraging tools like CSS Selectors, HTML DOM, and other methodologies [9], [10]. CSS Selectors have been chosen for their efficiency in bandwidth usage and memory optimization, making them a solution for web-based data analysis.

This study focuses on developing a chatbot integrated with web scraping capabilities to extract data from the Sci-Hub website, a platform often utilized by researchers to access scientific articles for free. Using Python and the BeautifulSoup library, this bot is designed to retrieve article information automatically, streamlining access for users. Sci-Hub, founded by Alexandra Elbakyan, provides access to over 88 million scientific articles, including those behind paywalls. Despite controversies surrounding copyright issues, Sci-Hub has become a symbol of resistance against restricted knowledge access, especially in developing countries.

The aim of this research is to develop a Telegram-based bot system that leverages web scraping to assist researchers in efficiently accessing scientific article data. This initiative is expected to contribute to the advancement of communication technology and promote broader dissemination of knowledge.

2.0 RELATED WORK

Several studies have explored the development and implementation of chatbots on the Telegram platform, showcasing the versatility of bots in addressing various needs through automation and user interaction. These works highlight the potential of combining programming, artificial intelligence, and web technologies to deliver efficient and user-friendly solutions.

Research by Avisyah et al. [11] developed a chatbot integrated with Telegram, leveraging artificial intelligence through Microsoft Azure to create a Question and Answer (QnA) system for educational purposes. The bot mimicked human-like interpersonal conversations by understanding the context of messages, functioning 24/7, including during holidays. This real-time response capability significantly reduced waiting times for users, showcasing how AI-powered chatbots can efficiently handle user queries. Meanwhile, [12] implemented a Telegram bot designed to retrieve information and movie schedules using the Hypertext Preprocessor (PHP) programming language. The bot successfully interacted with Telegram clients on both smartphones and computers. This study demonstrated the practical utility of Telegram bots in delivering specific content, such as entertainment-related information, to users.

The research by [13] utilizes Python-based web scraping to gather and update journal data from the SINTA website, simplifying the process for researchers by automating data collection and addressing challenges like manual searches and ranking analysis. This aims to streamline journal selection for publication. Besides that, Ferelestian et al. [14] created a Telegram-based student information chatbot using Wit.ai. The chatbot was designed to accelerate information delivery from websites to users, reducing the steps required to access information via traditional websites. Results showed that the chatbot, trained with Wit.ai's NLP services, could accurately process user queries and respond promptly.

Anam et al. [15] developed a Telegram chatbot using the Rasa framework to recommend tourist destinations in Semarang City. This research emphasized the importance of integrating Natural Language Processing (NLP) into chatbots to understand user preferences and deliver personalized recommendations. The chatbot significantly reduced search time while providing an engaging user experience.

The current research differentiates itself from previous studies by focusing on the unique functionality and massive data processing capability of the bot. While prior works primarily centered around specific system information or interactive QnA systems, this study extends the scope by integrating web scraping with a bot capable of delivering scientific articles and journals on a large scale. Unlike similar applications, such as Publish and Perish or Telegram-based Sci-Hub bots, the proposed system introduces enhanced features, including the ability to download multiple articles simultaneously.

Additionally, the bot offers opportunities for further development, such as implementing advanced search systems and multilingual support, which could make the application more accessible to a global audience. By combining the strengths of web scraping, Telegram API, and Python's robust libraries, this study contributes a novel approach to leveraging chatbots for academic and research purposes, addressing existing gaps in bot's functionality and usability.

3.0 RESEARCH METHODOLOGY

3.1 System Design

The system design phase focuses on determining how the proposed system addresses the identified problem for the research object. The system design in figure 1 provides an overview of what needs to be accomplished and how the Telegram bot operates within Telegram Messenger using the web scraping method. It outlines the framework for the system to be developed in this study, serving as a blueprint for implementation.

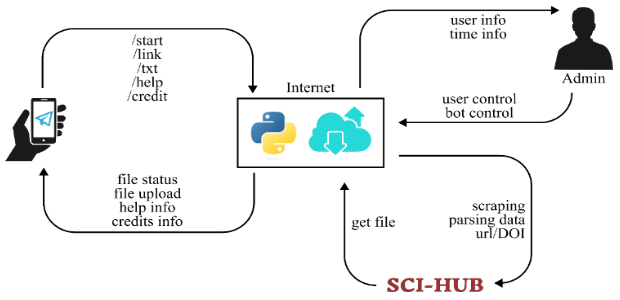


Figure 1: System Design

Figure 1 illustrates the interconnected processes within the system, starting with the user opening the Telegram application. This is followed by communication with the Telegram server, which is linked to the Sci-Hub website using the web scraping method (CSS Selector) implemented by the system.

Users send requests via the Telegram application installed on their devices, interacting with a bot created and configured through BotFather using specific commands. Commands are instructions provided by Telegram Messenger that allow developers to define the bot's functionality. The Telegram server receives the user's messages and forwards them to the bot server. The bot server, running a pre-designed program, processes the incoming messages to execute commands and provide accurate responses to users in the form of text or documents.

Each message acts as a command, influencing the nature of the response delivered to the user. The user can then interact further by following the instructions contained in the responses returned by the server. This iterative exchange ensures a smooth and responsive interaction process between the user and the Telegram bot.

3.2 System Implementation

In this phase, the system design was implemented in Python using Visual Studio. The developed system is a Telegram bot that employs web scraping methods, utilizing Python libraries. The implementation process began with developing the system design, including flowcharts and writing the program to be integrated into the system. The user interface was subsequently applied to align with the user-centric design.

After creating a Telegram account, users can access the Telegram bot specifically designed for the bidding system in this research. First, users search for the bot's ID, @ifunsil_bot, or its name, USA Downloader.

When using the Telegram bot as shown in figure 2, once the bot @ifunsil_bot is found, users are directed to an initial chat interface containing brief instructions on how to use the bot. Users can start the interaction by entering the /start command or pressing the start button on mobile devices. After initiating the conversation, the bot greets the user by name and provides guidance on using the available commands, such as downloading a single journal, downloading multiple journals, accessing bot usage information, and viewing credits. Each command is processed and responded to according to the input provided by the user.

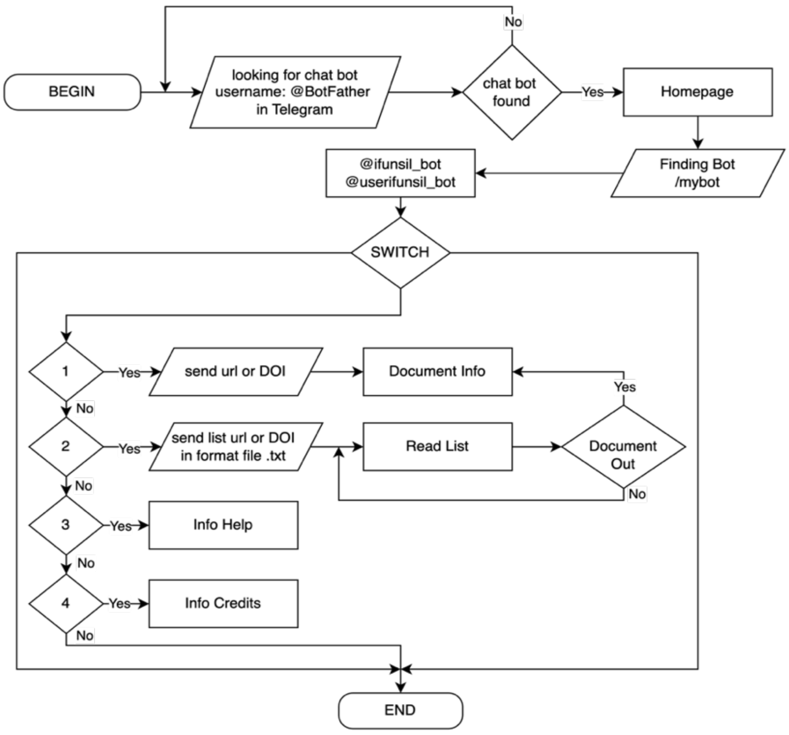


Figure 2: Flowchart for Using the Bot

3.3 System Testing

In this phase, several tests were conducted. The first test evaluated the functionality of the available features and ensured that the responses to user inputs were accurate. The second test measured the processing time during a request when a user sent a command to the Telegram bot via the server. This test involved timing the duration from the request initiation to the bot's response using a stopwatch, repeated 10 times for each response. The results of these tests provided data on the speed and responsiveness of the Telegram bot.

4.0 RESULT AND DISCUSSION

The program to be implemented aims to simplify the process for users to download scientific articles from Sci-Hub using the article's link or DOI. It will utilize web scraping techniques to retrieve the download link for articles from the Sci-Hub website. Currently, access to scientific articles is often restricted by paywalls or expensive subscriptions. Sci-Hub serves as an online repository providing free access to scientific journals and research articles that would typically require payment. This program enables users to access these articles easily without the need for payments or subscriptions.

The program is implemented using Python, utilizing libraries and modules such as requests for HTTP requests, BeautifulSoup for web scraping, and emoji for adding emoticons to user messages. Interaction with users is facilitated through the Telegram Bot API, where commands are provided for specific functions: /link is used for downloading a single article via a link or DOI, /txt allows multiple articles to be downloaded by submitting a .txt file containing links or DOIs, /help is made available for guidance, and /credits is provided to display details about the developers and academic advisors.

4.1 Implementasi Program Bot Chat

In the program implementation, the concept of web scraping will be utilized to extract the required information from the Sci-Hub webpage. The retrieved data will be processed to obtain the article download link, and the article will then be downloaded and saved in PDF format. This program implementation is expected to benefit researchers, students, and anyone in need of quick and free access to scientific articles. By using this program, users can save time and costs in accessing critical scientific resources essential for research and learning.

This ensures that the program is designed for efficient and user-friendly article retrieval. Next, the implementation of the program will be carried out using the Python programming language, with the previously mentioned libraries and modules being utilized.

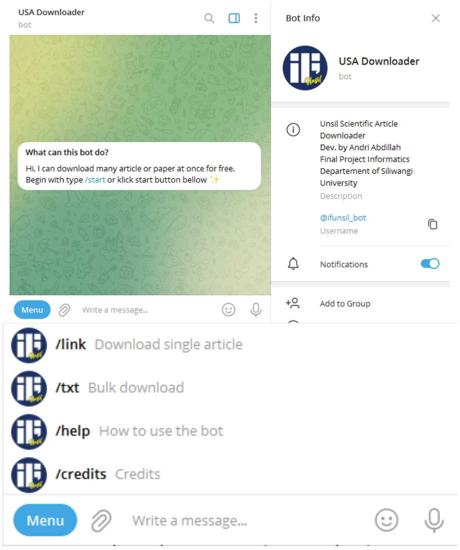
The pseudocode in table 1 illustrates the download_article() function, which accepts input in the form of a link or DOI of an article. This function performs an HTTP request to Sci-Hub using requests.get() and employs the BeautifulSoup library to scrape the webpage and locate the article's download link. If the article is found, the function downloads the article using another requests.get() call and saves it with a unique filename. In the main() function, the user is prompted to input the DOI or URL of the article, and then the download_article() function is called with the user input as its argument.

Table 1: Pseudocode Download Artikel

```
BEGIN
FUNCTION download_article (link_or_doi):
   IF link_or_doi.startswith("https://doi.org/"):
      doi <- link_or_doi.replace("https://doi.org/", "")
   ELSE:
      doi <- link_or_doi
   link <- 'https://sci-hub.se'
   res <- requests.get(link + '/' + doi, headers=header, timeout=30)
   soup <- BeautifulSoup(res.content, 'html.parser')
   content <- soup.find('embed')
   IF content IS NULL THEN
      PRINT("Article not found on Sci-Hub")
      RETURN
   END IF
   content <- content.get('src').replace('#navpanes=0&view=FitH', '')

   IF content.startswith('//zero.sci-hub.se') THEN
      download_url <- 'https:' + content
   ELSE
      download_url <- link + content
   END IF

   req <- requests.get (download_url, stream=True)
   file_name <- generate_unique_file_name() #Generate a unique file name to save the article

   WITH open (file_name, 'wb') AS file:
      file.write(req.content)

   PRINT("Article downloaded successfully!")
   RETURN

FUNCTION main():
   user_input <- input("Enter the article DOI or URL: ")
   download_article(user_input)

CALL main()
```

4.2 Implementasi Interface Bot Chat

A Telegram bot is created using BotFather by executing the /newbot command and providing the bot's name and ID. To access the bot, users can search for it within Telegram using its username via the platform's search engine or access it externally through the link https://t.me/ifunsil_bot.



USA Downloader
bot

What can this bot do?

Hi, I can download many article or paper at once for free.
Begin with type /start or klick start button bellow ✨

Bot Info

USA Downloader
bot

Unsil Scientific Article
Downloader
Dev. by Andri Abdillah
Final Project Informatics
Departement of Siliwangi
University
Description

@ifunsil_bot
Username

Notifications

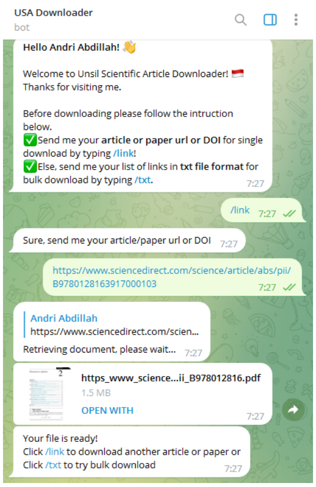
Add to Group

/link Download single article
/txt Bulk download
/help How to use the bot
/credits Credits

Figure 3: Bot Info and Commands on Telegram Desktop

The bot info and command as shown in figure 3 briefly outlines the capabilities of the Telegram bot "USA Downloader" and includes information about the bot's username, @ifunsil_bot, and its bio description. The registered commands are configured via BotFather, a Telegram bot designed to create and manage other bots. Users can access the list of available commands by tapping the menu button in the lower-left corner of the chat interface. These commands represent features that the bot can understand and respond to, with each sent command receiving an immediate response from the bot.

Testing chat demonstration in figure 4 was conducted by sending commands and utilizing the features available in the bot's menu. Commands in the chat bot always begin with a forward slash (/) and are processed based on the subsequent text. Available commands include: start, link, txt, help, and credits. Input in the form of text without a preceding forward slash is categorized into three types: links, DOIs, and random text. Any text sent immediately following the /link command is identified as either a link or DOI, while random text refers to input that is neither a command nor begins with a forward slash. The bot responds to unrecognized commands or random text with a message indicating that the input cannot be understood.


USA Downloader
bot

Hello Andri Abdillah! 👋

Welcome to Unsil Scientific Article Downloader! 🇮🇩
Thanks for visiting me.

Before downloading please follow the intruction below.
✅Send me your **article or paper url or DOI** for single download by typing /link!
✅Else, send me your list of links in **txt file format** for bulk download by typing /txt. 7:27

/link 7:27

Sure, send me your article/paper url or DOI 7:27

https://www.sciencedirect.com/science/article/abs/pii/B9780128163917000103 7:27

Andri Abdillah
https://www.sciencedirect.com/scien...
Retrieving document, please wait... 7:27

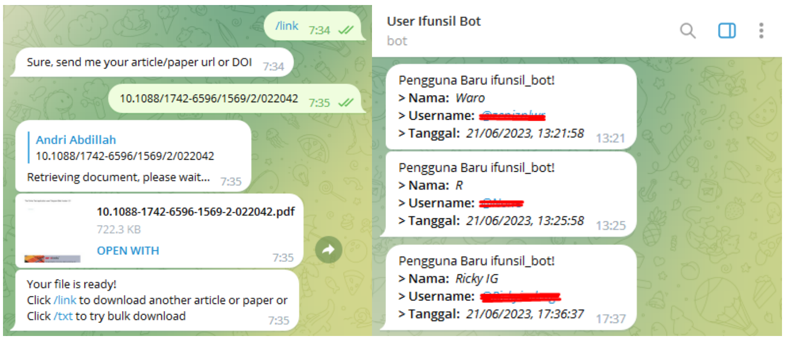
https_www_science...ii_B978012816.pdf
1.5 MB
OPEN WITH 7:27

Your file is ready!
Click /link to download another article or paper or
Click /txt to try bulk download 7:27

Figure 4: Chat Demonstration

4.3 Testing Result

The bot's download feature is divided into two categories: single article downloads, and bulk article download. The single download feature allows the bot to download one article at a time using the /link command, followed by submitting the URL or DOI of the desired article as shown in figure 5.



/link 7:34

Sure, send me your article/paper url or DOI 7:34

10.1088/1742-6596/1569/2/022042 7:35

Andri Abdillah
10.1088/1742-6596/1569/2/022042
Retrieving document, please wait... 7:35

10.1088-1742-6596-1569-2-022042.pdf
722.3 KB
OPEN WITH 7:35

Your file is ready!
Click /link to download another article or paper or
Click /txt to try bulk download 7:35

User Ifunsil Bot
bot

Pengguna Baru ifunsil_bot!
> Nama: *Waro*
> Username: [redacted]
> Tanggal: *21/06/2023, 13:21:58* 13:21

Pengguna Baru ifunsil_bot!
> Nama: *R*
> Username: [redacted]
> Tanggal: *21/06/2023, 13:25:58* 13:25

Pengguna Baru ifunsil_bot!
> Nama: *Ricky IG*
> Username: [redacted]
> Tanggal: *21/06/2023, 17:36:37* 17:37

Figure 5: Single Download Feature and User Information Notification

Table 2 shown as bot chat response time. From the results of 100 test trials for each response type, it was found that the bot's response time to user messages in text form was highly responsive, with an average response time of 1.06 seconds for content with an average length of 200 characters. In contrast, responses that included files had a longer response time, averaging 7.97 seconds for content with an average length of 171 characters and files in PDF format with an average size of 977 kilobytes.

Table 2: Bot Chat Response Time

No	Input	Respond		Number of orders received	Average Time
		text (char)	file (kb)		
1	/strart	318	-	1	0,944
2	/link	43	-	1	0,856
3	url	134	1580	3	9,593
4	doi	134	503,2	3	7,482
5	/txt	58	-	1	1,179
6	.txt file	588	2805	10	22,793
7	url/doi salah	99	-	1	1,739
8	pesan acak	62	-	1	1,084
9	/help	491	-	1	0,835
10	/credits	307	-	1	0,844

5.0 CONCLUSION

The implemented features function effectively, allowing the bot to download journal articles based on user requests. Using the /link command, users can download articles by providing a URL or DOI, while the /txt command enables the download of multiple URLs and DOIs in bulk through a *.txt file. The response time for the Telegram bot, which utilizes web scraping methods, averages 1.06 seconds for content with a length of 200 characters. Responses involving files take longer, with an average response time of 7.97 seconds for content with a length of 171 characters and files in PDF format averaging 977 kilobytes in size.

ACKNOWLEDGMENTS

We would like to express our deepest gratitude to the Department of Informatics, Siliwangi University for their support and guidance throughout this research. Special thanks to all for their valuable effort.

REFERENCES

[1] H. Elmi, A. Ambiyar, Y. Huda, and D. Novaliendry, “The Role of Information and Communication Technology in Interactive Learning,” *Jurnal SAINTIKOM (Jurnal Sains Manajemen Informatika dan Komputer)*, vol. 23, no. 1, p. 193, Feb. 2024. DOI: 10.53513/jis.v23i1.9549

[2] A. Haleem, M. Javaid, M. A. Qadri, and R. Suman, “Understanding the role of digital technologies in education: A review,” *Sustainable Operations and Computers*, vol. 3, pp. 275–285, 2022. DOI: 10.1016/j.susoc.2022.05.004

[3] K. A. Persichitte and A. Suparman, *Educational Technology to Improve Quality and Access on a Global Scale*. Cham: Springer International Publishing, 2018. DOI: 10.1007/978-3-319-66227-5

[4] S. Nawaz, “Rethinking classifications and metrics for problematic smartphone use and dependence: Addressing the call for reassessment,” *Computers in Human Behavior Reports*, vol. 12, p. 100327, Dec. 2023. DOI: 10.1016/j.chbr.2023.100327

[5] L. Thomas and S. Bhat, “A Comprehensive Overview of Telegram Services - A Case Study,” *International Journal of Case Studies in Business, IT, and Education*, pp. 288–301, May 2022. DOI: 10.47992/IJCSBE.2581.6942.0165

[6] I. R. Kalantarov and D. Y. Volkov, “Development of A Telegram-Bot For Automating The Educational Process,” *Current Problems Of Teaching Mathematics At Technical University*, vol. 10, pp. 53–56, 2023. DOI: 10.25206/2307-5430-2023-10-53-56

[7] G. F. Avisyah, I. J. Putra, and S. S. Hidayat, “Open Artificial Intelligence Analysis using ChatGPT Integrated with Telegram Bot,” *Jurnal ELTIKOM*, vol. 7, no. 1, pp. 60–66, Jun. 2023. DOI: 10.31961/eltikom.v7i1.724

[8] A. Abodayeh, R. Hejazi, W. Najjar, L. Shihadeh, and R. Latif, “Web Scraping for Data Analytics: A BeautifulSoup Implementation,” in *2023 Sixth International Conference of Women in Data Science at Prince Sultan University (WiDS PSU)*, 2023, pp. 65–69. DOI: 10.1109/WiDS-PSU57071.2023.00025

[9] C. Lotfi, S. Srinivasan, M. Ertz, and I. Latrous, “Web Scraping Techniques and Applications: A Literature Review,” in *SCRS Conference Proceedings on Intelligent Systems*, Soft Computing Research Society, 2021, pp. 381–394. DOI: 10.52458/978-93-91842-08-6-38

[10] 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

[11] G. F. Avisyah, I. J. Putra, and S. S. Hidayat, “Open Artificial Intelligence Analysis using ChatGPT Integrated with Telegram Bot,” *Jurnal ELTIKOM*, vol. 7, no. 1, pp. 60–66, Jun. 2023. DOI: 10.31961/eltikom.v7i1.724

[12] S. Bahrurozi, A. Asroni, and C. Damarjati, “Advanced Development of a Prayer Schedule Bot Application on Telegram Using PHP,” *Emerging Information Science and Technology*, vol. 1, no. 4, pp. 144–148, Oct. 2022. DOI: 10.18196/eist.v1i4.16596

[13] 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

[14] V. Joey Ferelestian, B. Susanto, and I. K. D. Senapartha, “Pengembangan Telegram Chatbot Informasi Mahasiswa Menggunakan Wit.ai,” *Jurnal Terapan Teknologi Informasi*, vol. 7, no. 2, pp. 89–97, Oct. 2023. DOI: 10.21460/jutei.2023.72.257

[15] Farid Asroful Anam, “Telegram Chatbot Implementation Using Rasa Framework to Recommend Tourism in Semarang City,” *Jurnal Ilmiah Komputasi*, vol. 23, no. 1, Mar. 2024. DOI: 10.32409/jikstik.23.1.3569