KLIK: Kajian Ilmiah Informatika dan Komputer
ISSN 2723-3898 (Media Online)
Vol 4, No 3, December 2023, Hal 1829-1837
DOI 10.30865/klik.v4i3.1547
https://djournals.com/klik

Derivative Words Scraping of Every Quranic Root Word from the Quran Corpus Web using Python to Support the Quranpedia Project

Idzhari Syaeful Ma'mun, Eko Darwiyanto, Moch. Arif Bijaksana*

Fakultas Informatika, Program Studi S1 Rekayasa Perangkat Lunak, Telkom University, Bandung, Indonesia
Email: [1]idzharism@student.telkomuniversity.ac.id, [2]ekodarwiyanto@telkomuniversiy.ac.id, [3,*] arifbijaksana@telkomuniversiy.ac.id
Email Penulis Korespondensi: idzharism@student.telkomuniversity.ac.id

Abstract−The Qur'an, as a guide to life for Muslims, has given birth to various disciplines such as tafsir science, fiqh science, hadith science, nahwu science, and balaghah science. However, the limited number of websites on learning and understanding the Qur'an is a problem that can hinder Muslims from exploring the contents of the Qur'an. To overcome this problem, the Quranpedia project was initiated. Quranpedia is a web-based application designed to resemble Wikipedia in providing in-depth explanations of derivative words in the Qur'an. Using the "Scraping" technique, Quranpedia collects data from various sources to provide a comprehensive explanation of nouns in the Qur'an and Hadith. One of the main challenges in this project was to find the common root of nouns in the Qur'an and hadith. To overcome this challenge, a method was used to transform words from sentences to their root words. Thus, Quranpedia can have the ability to look up the root word of a noun. This allows users to have a better understanding of derivative words in the Qur'ān and how they are used in different contexts. The objective of this research is to create a derivative word scraping program that scrapes all derivative words in the Quran from the Corpus Quran web accurately. The problem discussed in this research covers both how one can scrape derivative words of each root word in the Quran from the Corpus Quran web and whether the data scraped from the web is complete and accurate. The method to ensure that these problems are solved includes using the Python programming language to create the program and then testing the program itself. The interim results achieved is whether the data is complete or not.

Keywords: Scraping; Quranpedia; Quran; Derivative Words

1. INTRODUCTION

The Quran, revered as the holy book of Islam, is considered by its followers to be the complete and perfect word of Allah which was revealed onto the Prophet Muhammad pbuh. through the angel Gabriel. [1] It is not merely a religious text containing rules and regulations, but also a comprehensive guide to life. The Qur'an, as a guide to life for Muslims, has given birth to various disciplines such as tafsir science, fiqh science, hadith science, nahwu science, and balaghah science. [2] The Quran pays attention to the minutest details, making it a source of knowledge and wisdom for its followers. [3] The Quran, as a religious text, serves not just as a guide for spiritual and moral conduct, but also as a comprehensive manual for life. It addresses a wide range of topics, from the most profound philosophical questions to the most mundane aspects of daily life. This makes the Quran a rich and complex text, full of nuances and subtleties that can be challenging to understand without proper guidance. Islam's conceptualization of existence begins with God's command to create nature, symbolized by the kalām (word): "*kun fa yakūn*" (be, and it is). [4]

The Holy Quran was revealed in the Arabic language. The standard version of Arabic is derived from Classical Arabic, which has also been used as the language of literature and the language of Islamic worship since around the sixth century. The Arabic language also plays an important role in the development of the Islamic civilization. [5] Arabic is still considered by most students as a language that is difficult to learn, even seen as a field of study that is not liked. Likewise, in terms of the implementation of teaching, many problems are faced, starting from the elementary level to the university level. [6] The root word in the Arabic language may also be referred to as a *Masdar*. A *Masdar* is a word that indicates an event but the word itself is not bound by time. [7] One may look up derivative words based on the available *Masdar*s. The Arabic language derivation practically refers to producing a new word from another word; both words have the same root and the same general lexical meaning but are not morphologically the same. [8] When it comes to the Arabic language, there are rules that shape the Arabic grammar that are used to derive and analyze syntaxes of Arabic sentences. [9]

In the digital age, Wikipedia has emerged as a popular platform that provides encyclopedic discussions on a wide array of topics. Each topic is presented with the latest research findings, making it an excellent reference for beginners seeking to understand a particular subject. However, when it comes to religious texts like the Quran, there is a noticeable gap.

This gap led to the conception of "Quranpedia", a hypothetical platform where every noun in the Quran is explained in detail. The explanations would include relevant verses from the Quran, related hadiths from Kutubus Sittah, and information from Wikipedia. It's important to note that at the time of this research, Quranpedia was not yet created, hence no web address could be provided.

This is where Quranpedia comes in. By providing clear, comprehensive explanations of every noun in the Quran, Quranpedia aims to make the Quran more accessible and understandable to a wider audience. The explanations would draw from a variety of sources, including relevant verses from the Quran, related hadiths from Kutubus Sittah, and information from Wikipedia. This multi-source approach ensures that the explanations are as accurate and comprehensive as possible.

The creation of Quranpedia would involve the use of "Web Scraping", a technique used to extract data from the World Wide Web (WWW) and store it into a file system or database for later retrieval or analysis. [10], [11] Web data is

typically retrieved using the Hypertext Transfer Protocol (HTTP) or through a web browser. By employing this technique, Quranpedia could gather data from various sources to provide a comprehensive explanation of nouns in the Quran and Hadith. The use of web scraping techniques is crucial to the creation of Quranpedia. Web scraping is a powerful tool that allows for the extraction of data from the World Wide Web (WWW) and its storage into a file system or database for later retrieval or analysis. This technique would enable Quranpedia to gather data from various sources, providing a comprehensive explanation of nouns in the Quran and Hadith.

The choice of encyclopedia topics is a critical aspect of Quranpedia. These topics, which can range from nouns, verbs, adjectives, phrases, or acronyms to specialized terms and words in foreign languages, are chosen for their relevance and significance. The aim is to provide readers with a clear and comprehensive explanation of these topics.

Encyclopedia topics are usually chosen based on their significance to provide readers with a clear and comprehensive explanation. [12] These topics can range from nouns, verbs, adjectives, phrases, or acronyms to specialized terms and words in foreign languages. The choice of topic is crucial as it determines the scope and depth of the information provided.

he idea of Quranpedia was born out of the need to understand the science of Islam better, which is the religion with the most followers in the author's home country, Indonesia. [13] This topic was deemed suitable for a final project as it involves creating and researching web pages and developing programs to extract specific data from a website. This aligns well with the study program undertaken by the author.

The importance of Arabic root words in this project cannot be overstated. Arabic root words can help decipher the meaning of each word contained in the Quran, which is written in Arabic. This would greatly enhance the understanding of the Quran and contribute to the overall effectiveness of Quranpedia. Arabic root words play a significant role in this project as they can help decipher the meaning of each word contained in the Quran, which is written in Arabic. [13] The ideal scenario would be to retrieve and store the data contained in the benchmark website, compared to the current condition which still relies on databases whose sources are still from external sources to display data on the web.

The benchmark website that will be used is the Corpus Quran web. The Corpus Quran web is an annotated linguistic resource which shows the Arabic grammar, syntax, and morphology for each word in the Holy Quran. [14]

The reason why this research was done by writing a scraping program using Python, is because the customization of the program created can be freely adjusted accordingly to the research's needs. Compared to other methods, such as using standalone software, it could generate unnecessary data and information which are irrelevant to the research conducted. [11]

Some of the previously conducted research that is related to this research is as follows. The first one is the teaching of the Arabic language, being there are a multitude of problems experienced, starting from the elementary level up to the higher level of education. [6] Next up is the research on derivative words. Derivation in Arabic practically refers to creating new words from other words, with both words having the same root word and the same general lexical meaning, though morphologically different. There exist certain rules in the Arabic language that are used to derive and analyze syntaxes in Arabic sentences. [8], [9], [15] Last one is the research about web scraping. Web scraping or web crawling refers to the procedure of automatically retrieving data from websites using software. It is a very important process in fields like business intelligence in the modern era. Web scraping is a technology that allows us to extract structured data from text such as HTML files. [10], [11]

2. RESEARCH METHODOLOGY

2.1 Software Development Method

To develop the application for this research, a development method is necessary. The flow of the method is divided into seven stages. Such stages are shown in Figure 1 below.

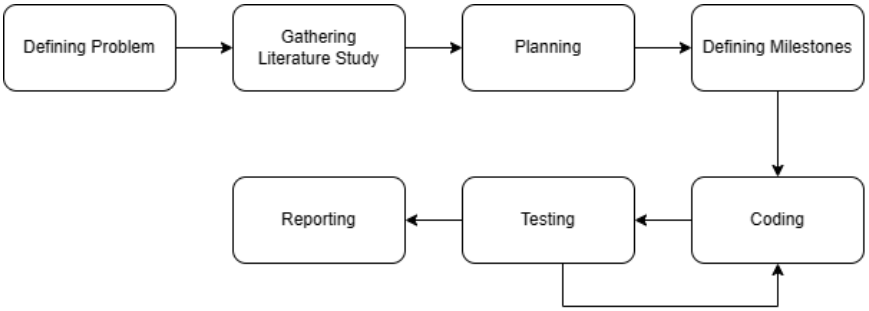

Figure 1. Software Development Stages Flowchart

The flowchart shown above contains the seven stages of the development of the program created. They consist of:

a. Defining Problem, which is listing all the problems faced and how to solve them.
b. Gathering Literature Study, reading and collecting previous studies that relate to the current research to help solve the problems faced.

c. Planning, after the literature studies are gathered, plan out how the research may be done.
d. Defining Milestones, define the milestones in research to keep tabs on progress.
e. Coding, apply the logic for the program accordingly to reach the intended outcome. The language used for this stage is Python. Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. The high-level data structures built into it, combined with dynamic typing and dynamic binding, make it very attractive for Rapid Application Development, as well as for use as a scripting language or glue language to connect components together. Python's simple, easy-to-learn syntax emphasizes readability and thus reduces program maintenance costs. Python supports modules and packages, which encourages program modularity and code reuse. The Python interpreter and extensive standard library are available in source or binary form at no cost for all major platforms and can be freely distributed. [16] Compared to other programming languages, Python's syntax is designed for readability and has many similarities to English with influences from mathematics. [17]
f. Testing, test the code created and see whether it can produce the intended outcome or not. The testing metric used is called Cyclomatic Complexity. Cyclomatic complexity is a software metric used to measure code complexity. This metric measures independent paths through the source code. The point of this requirement is to minimize risk, minimize testing, and increase reliability associated with the code of mission-critical software components. [18]
g. Reporting, report how the code performed based on the testing results.

2.2 Program Flowchart

The process of data extraction from the web, specifically from the benchmark web Corpus Quran, is a multi-step procedure that involves several stages. Below is Figure 2 which illustrates the flow of the program.

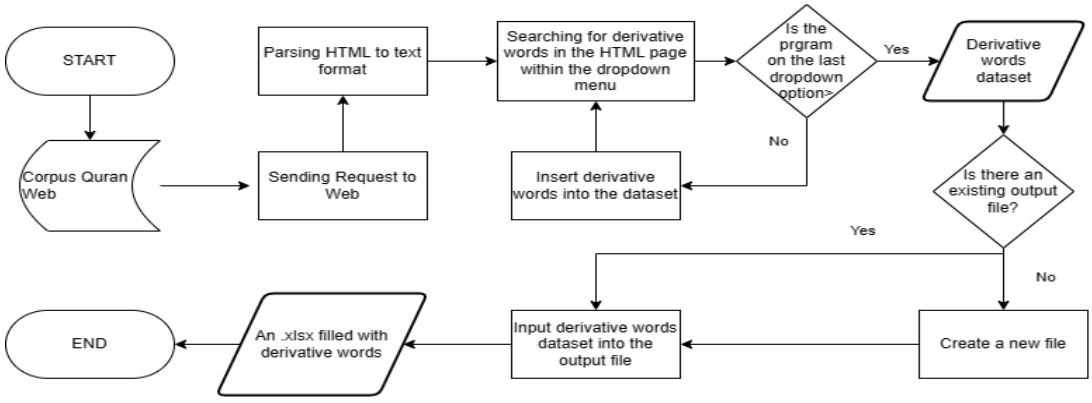

Figure 2. Program Flowchart

The stages that the program goes through to scrape the data goes as follows:

a. Initially, the program makes a request to establish a connection with the benchmark web. This is a crucial first step as it allows the program to access the data contained on the web page.
b. Once the program has successfully made the request, it proceeds to retrieve the HTML from the web page. The HTML, which is the backbone of any web page, contains all the information displayed on the page.
c. However, this raw data is not immediately usable by the program. Therefore, the program performs a process known as parsing on HTML. Parsing transforms the HTML into a format that can be easily read and manipulated by the program.
d. With the parsed HTML, the program can now search for derivative words within it. These derivative words are distinguished by their green color and are wrapped using the <span class="at> tag. Armed with this information, the program searches for all instances of this tag in HTML.
e. It continues to loop through each <option> tag contained within the <select> tag, which represents a dropdown on the benchmark web Corpus Quran.
f. After the program completes the looping process to find the derivative words, the data obtained from the benchmark web is inserted into a dataset available within the Python program. This dataset serves a crucial function as it collects the derivative words before they are inserted into the output file.
g. Before proceeding to the next step, the program checks if an output file has already been created. If no output file exists, the program creates a new one. However, if an output file already exists, the program proceeds to the data input stage.
h. The dataset that has been collected in the program is then processed using the libraries available from the Python programming language.
i. This data is output in the form of a file with an .xlsx extension. This type of file can be opened using applications such as Microsoft Excel, making it accessible for further analysis and manipulation.

In conclusion, the process of extracting data from the web, specifically from the benchmark web Corpus Quran, is a complex procedure that involves several stages. From making a request to connect with the web, retrieving and parsing the HTML, searching for derivative words, collecting the data into a dataset, checking for an existing output file, to finally

processing the data and outputting it into a file, each step plays a crucial role in ensuring the successful extraction of data. This process, powered by the Python programming language and its libraries, demonstrates the power and potential of web scraping as a tool for data extraction and analysis. With this technique, valuable insights can be gleaned from the vast amount of data available on the web, contributing to various fields of study and research.

2.3 Testing

The testing for this research will be conducted using two different methods. The first one would be to check the accuracy of the output that will be generated by the program by comparing it to the Corpus Quran web as the benchmark web. The second one would be to measure the complexity of the program itself by using the Cyclomatic Complexity matrix. Cyclomatic complexity is a software metric used to measure code complexity. This metric measures independent paths through the source code. The point of this requirement is to minimize risk, minimize testing, and increase reliability associated with the code of mission-critical software components. [18], [19]

3. RESULT AND DISCUSSION

The primary focus of this research encompasses both the coding and testing phases, and the results derived from them. These phases are critical components of any software development process. In this case, they are used to implement the necessary logic, generate the output by the program, and measure the complexity of the code.

The coding phase involves implementing the logic required to scrape derivative words data from the Corpus Quran web. This includes designing the program's structure, writing the code, handling potential errors, and ensuring the code is maintainable and efficient. The success of this stage is critical for the overall success of the research.

Following the coding phase, the testing phase is conducted with the purpose of determining whether the output generated by the program aligns with the expected results. This is important because the accuracy of the output is a direct measure of the program's effectiveness. If the output is accurate, it means that the program is functioning as intended. On the other hand, if the output is not accurate, it indicates that there may be errors or bugs in the code that need to be fixed in future research.

In addition to testing the accuracy of the output, the testing phase also involves assessing the complexity of the code. Code complexity can impact the efficiency and maintainability of the program. Therefore, it is important to keep the code as simple as possible while still achieving the desired functionality.

The process of testing involves creating an output file containing derivative words data scraped from the Corpus Quran web. The results obtained from the testing phase provide valuable information into the program's performance and effectiveness, contributing significantly to the overall results of the research.

In conclusion, both the coding and testing phases are integral parts of this research. They function to implement the necessary logic, validate the accuracy of the program's output, and measure the complexity of the code. Therefore, both coding and testing are essential to ensure the reliability and validity of the research results.

3.1 Coding

The coding stage is a crucial part of this research, where the theoretical concepts and logic are transformed into a functional program. This stage involves the implementation of the logic required to scrape derivative words data from the Corpus Quran web. The process begins with the design of the program's structure, which includes defining the necessary functions and classes.

The next step is the actual coding, where the functions and classes are filled with the appropriate logic. This includes writing code to access the web page, parse the HTML to locate the data, extract the data, and finally write the data to an output file. Each line of code is written with careful consideration to ensure it aligns with the overall logic of the program.

Error handling is also an integral part of the coding stage. This involves writing code to handle potential errors or exceptions that may occur during the execution of the program. This is important to ensure the program can recover gracefully from errors and continue its operation.

Moreover, the code is written with a focus on maintainability and efficiency. This means keeping the code as simple and readable as possible, which can help in future modifications and debugging. Comments are also added to the code to explain the functionality of each part, which can be beneficial for other researchers who might work on this project in the future.

In conclusion, the coding stage is where the logic needed for the program is implemented. It involves several steps and considerations to ensure the program is effective, efficient, and maintainable. The success of this stage is critical for the overall success of the research.

3.2 Testing

The testing stage is divided into three separate stages. Those stages include measuring the program's output file accuracy, measuring the program's runtime, and measuring the program's complexity.

3.2.1 Program's output file accuracy

The first test is conducted to see how accurate the output file of the program. This will be measured by comparing the number of words that are listed on the benchmark web with the number of words in the output file of the program. The two compared numbers are then converted to a percentage format to produce more sensible data. The results of the accuracy test of the program are shown below in Table 1.

Table 1. Data Accuracy Comparison of The Corpus Quran Web Against the Program's Output

No.	Arabic letter	Number of words		Percentage
		Corpus Quran Web	Program's output	
1	أ	209	206	98.56%
2	ب	257	257	100%
3	ت	50	50	100%
4	ث	59	59	100%
5	ج	157	157	100%
6	ح	296	296	100%
7	خ	222	222	100%
8	د	107	107	100%
9	ذ	60	60	100%
10	ر	253	253	100%
11	ز	86	86	100%
12	س	308	308	100%
13	ش	156	156	100%
14	ص	191	191	100%
15	ض	66	66	100%
16	ط	112	112	100%
17	ظ	36	36	100%
18	ع	337	337	100%
19	غ	139	139	100%
20	ف	201	201	100%
21	ق	260	260	100%
22	ك	173	173	100%
23	ل	116	116	100%
24	م	181	181	100%
25	ن	296	296	100%
26	ه	86	86	100%
27	و	235	235	100%
28	ي	30	30	100%
		Average		99.95%

The table above displays the whole accuracy of the program that was made. As accurate as the program may be, when processing the first Arabic letter, it misses the first few words. This may be caused by a bug that made the program ignore a certain condition that may happen when the URL of the HTML page is not fully recognized by the program. Some of the missed words include آزَر, آدَم, and أَبَاريق. A deeper analysis shows that these three words appear as the very first three data in the Corpus Quran web, which hints at the possibility of the program not recognizing them as form of derivative words that should have been scraped, especially since the query string of the pages, also known as option value, that host these words are unique, being 'A%5Edam', 'A%5Ezar', and '%3EabaAriyq' respectively unlike any other words which are just simple three letter words such as 'Abb' and 'Abd'. Below is Figure 3, which contains a screenshot of the Corpus Quran web page's HTML structure which was inspected using an internet browser's inspect element function.

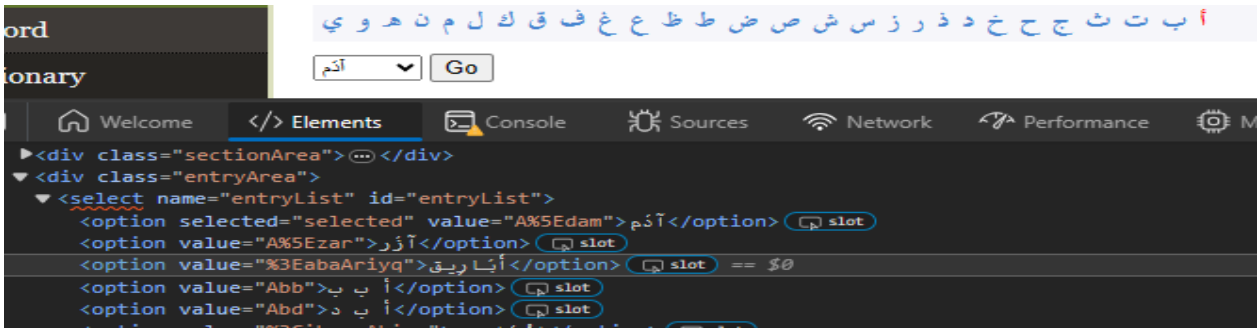

Figure 3. A screenshot of the Corpus Quran web page's HTML structure

The full data of the program's output is accessible on this link in the form of a table. https://drive.google.com/file/d/1gs_PtFOAS5-XloVyOtjYNGekS5cW5C-s/view

3.2.2 Program Run Time

After testing the program's accuracy in extracting the derivative words from the Corpus Quran web, the program's processing time is also tested afterwards. This was done to see how efficient the program is at processing the large amount of data from the Corpus Quran web. Below is Table 2, which shows the amount of time consumed to process each Arabic letter in the Corpus Quran web, as well as the total amount of time consumed processing all the Arabic letters.

Table 2. Processing Time for Each Letter

No.	Arabic letter	Number of root words	Processing time per letter
1	أ	87	48 Second(s)
2	ب ـبـ	83	40 Second(s)
3	ت ـتـ	20	7 Second(s)
4	ث ـثـ	21	8 Second(s)
5	ج ـجـ	69	24 Second(s)
6	ح ـحـ	98	34 Second(s)
7	خ ـخـ	71	24 Second(s)
8	د ـد	46	16 Second(s)
9	ذ ـذ	22	8 Second(s)
10	ر ـر	87	31 Second(s)
11	ز ـز	39	14 Second(s)
12	س ـسـ	105	38 Second(s)
13	ش ـشـ	63	23 Second(s)
14	ص ـصـ	63	22 Second(s)
15	ض ـضـ	25	9 Second(s)
16	ط ـطـ	36	12 Second(s)
17	ظ ـظـ	7	3 Second(s)
18	ع ـعـ	102	39 Second(s)
19	غ ـغـ	50	18 Second(s)
20	ف ـفـ	72	26 Second(s)
21	ق ـقـ	80	33 Second(s)
22	ك ـكـ	61	24 Second(s)
23	ل ـلـ	55	22 Second(s)
24	م ـمـ	67	24 Second(s)
25	ن ـنـ	105	44 Second(s)
26	ه ـهـ	39	14 Second(s)
27	و ـو	77	29 Second(s)
28	ي ـيـ	11	11 Second(s)
	Total processing time		10 Minutes 45 Second

The program runs for roughly about 10 minutes and 42 seconds. The time spent processing each of the letters is rounded to the nearest whole number, which is why the writer uses words such as 'about' and 'roughly' when describing the amount of time taken to process the words. The processing speed may also vary depending on the connection to the web and the hardware used to execute the program.

3.2.3 Console Output

The program also prints out to the console to give more context to the progress of the scraping process. This helps to track how much of the data has been processed and how much time was consumed on each "milestone". Below are a few screenshots of the console output that the program prints out to the console. The first one is Figure 4, which shows the output console of the program while processing the list_sectionArea A.

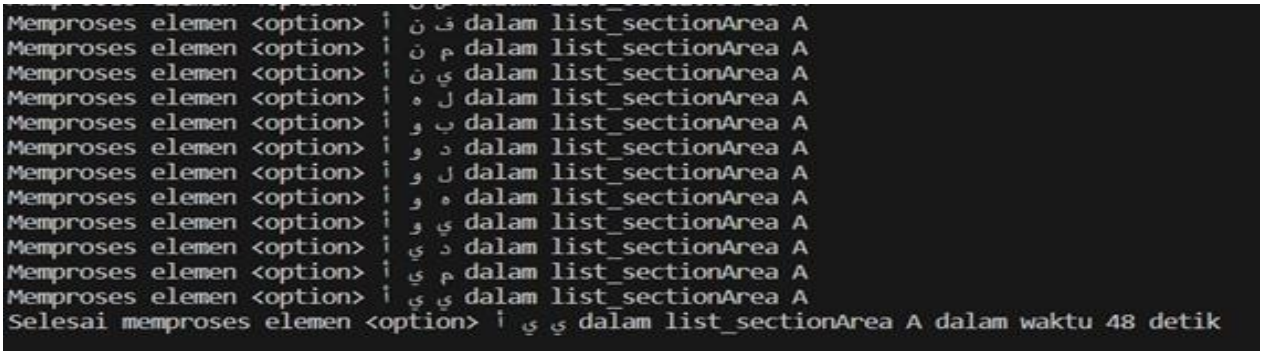

Figure 4. Program's console output while processing list_sectionArea A

The figure above shows the program takes about 48 seconds of time to process the list_sectionArea 'A'. It takes 48 seconds to process, which according to Table 2 is the longest, because of the amount of data stored in the list_sectionArea 'A'. The list_sectionArea 'A' stores the largest amount of data, which means that most derivative words in the Quran's root words start with the letter Alif, which in this case is represented using the alphabetical letter 'A'.The second one is Figure 5, which shows the output console of the program while processing the list_sectionArea b and is shown below.

```
Memproses elemen <option> ب ه ج dalam list_sectionArea b
Memproses elemen <option> ب ه ل dalam list_sectionArea b
Memproses elemen <option> ب ه م dalam list_sectionArea b
Memproses elemen <option> ب و أ dalam list_sectionArea b
Memproses elemen <option> ب و ب dalam list_sectionArea b
Memproses elemen <option> ب و ر dalam list_sectionArea b
Memproses elemen <option> ب و ل dalam list_sectionArea b
Memproses elemen <option> ب ي ت dalam list_sectionArea b
Memproses elemen <option> ب ي د dalam list_sectionArea b
Memproses elemen <option> ب ي ض dalam list_sectionArea b
Memproses elemen <option> ب ي ع dalam list_sectionArea b
Memproses elemen <option> ب ي ن dalam list_sectionArea b
Selesai memproses elemen <option> ب ي ن dalam list_sectionArea b dalam waktu 40 detik
```

Figure 5. Program's console output while processing list_sectionArea b

The figure above shows the program takes about 40 seconds of time to process the list_sectionArea 'A', which is slightly less time to process than the previous figure. This is because there are fewer derivative words in the Quran that start with the letter Ba, which in this case is represented with the alphabetical letter 'b'. The third one is Figure 6, which, as shown below, shows the output console of the program while processing the list_sectionArea y.

```
Memproses item y dari list_sectionArea...
Memproses elemen <option> ي أ س dalam list_sectionArea y
Memproses elemen <option> ي ب س dalam list_sectionArea y
Memproses elemen <option> ي ت م dalam list_sectionArea y
Memproses elemen <option> ي د ي dalam list_sectionArea y
Memproses elemen <option> ي س ر dalam list_sectionArea y
Memproses elemen <option> ي ق ظ dalam list_sectionArea y
Memproses elemen <option> ي ق ن dalam list_sectionArea y
Memproses elemen <option> ي م م dalam list_sectionArea y
Memproses elemen <option> ي م ن dalam list_sectionArea y
Memproses elemen <option> ي ن ع dalam list_sectionArea y
Memproses elemen <option> ي و م dalam list_sectionArea y
Selesai memproses elemen <option> ي و م dalam list_sectionArea y dalam waktu 11 detik

Total waktu eksekusi program adalah 17 menit dan 35 detik
Data Berhasil Ditulis Kedalam D:\TA\UPLOAD\BARU\SCRAPPING\SCRAPPING_UPDATE_ADA TIMER\output.xlsx
PS D:\TA\UPLOAD\BARU\SCRAPPING\SCRAPPING_UPDATE_ADA TIMER>
```

Figure 6. Program's console output while processing list_sectionArea y

The figure above shows the program takes about 11 seconds of time to process the list_section area 'y'. This is because in the Quran, there are significantly fewer derivative words in the Arabic language that start with the letter Ya, which in this case is represented using the alphabetical letter 'y'.

3.2.4 Cyclomatic Complexity

From the results of the conducted analysis, an analysis is then made from the testing stage that was done. Below is Figure 7 which is a graph illustrating the Cyclomatic Complexity of the program.

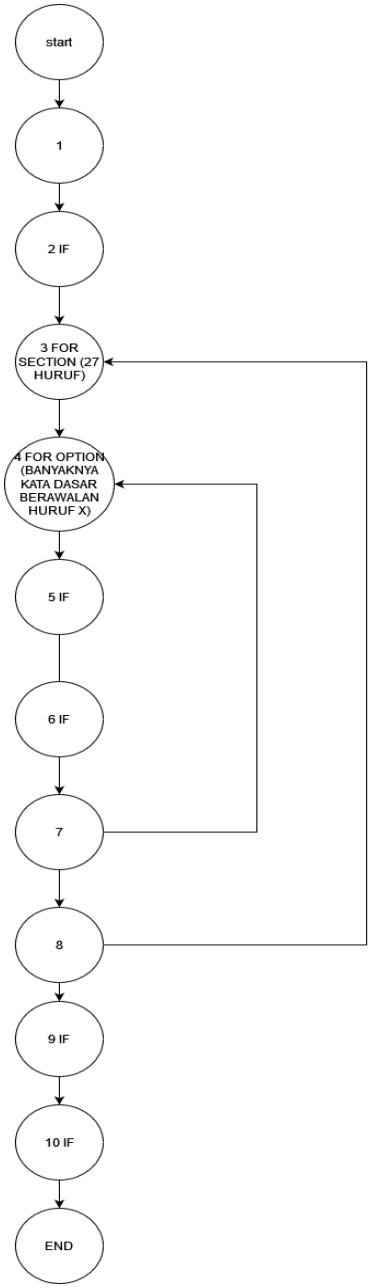

Figure 7. Cyclomatic Complexity Graph of the Program

The graph provided above serves as a visual representation of the operations performed within the code. Each operation is symbolized by a node, and there are a total of 12 distinct nodes, inclusive of the start and end nodes. The nodes are interconnected by 13 edges, signifying the flow and relationship between different operations.

The concept of cyclomatic complexity comes into play here. Cyclomatic complexity is a quantitative measure of the complexity of a program. It is computed using the control flow graph of the program, where the nodes represent the operations, and the edges represent the flow between these operations.

In the context of this program, the cyclomatic complexity matrix's value is derived from the number of enclosed areas plus one. From the given graph, it is observed that there are two enclosed areas. Therefore, the value of the enclosed areas is equal to 2.

With this value, the cyclomatic complexity of the program can be calculated using the formula:

$$Number\ of\ enclosed\ areas\ +\ 1\ =\ Cyclomatic\ Complexity \qquad (1)$$

Substituting the value of the enclosed areas into the formula gives:

$$2 + 1 = 3$$

This implies that the cyclomatic complexity value of this program is 3.

The cyclomatic complexity value provides information of the program's complexity. A lower cyclomatic complexity may mean that the program is relatively simple, with fewer paths through the code. On the other hand, a

higher cyclomatic complexity indicates a more complex program with more paths. This could make the program more difficult to understand, maintain, and modify.

In this case, a cyclomatic complexity value of 3 suggests that the program is relatively simple and straightforward. [20] This could be advantageous in terms of maintainability and understandability of the code. However, it is also essential to consider other factors such as the program's functionality and the efficiency of the code.

In conclusion, cyclomatic complexity is a useful metric for assessing the complexity of a program. It provides a measure that can help in understanding the program better, improving the design, and making more informed decisions when modifying the code. In the context of this program, the cyclomatic complexity value of 3 suggests a relatively simple and straightforward program. However, it is always important to consider this value with other factors and metrics to get an understanding of the program's complexity.

4. CONCLUSION

The conclusion that can be drawn from this research is that the scraping of derivative words of each Qur'anic root word from the Corpus Quran web can be done using the Python programming language and the results of the scraping program data from the web are quite complete and accurate. The test results show that the program created has a Cyclomatic Complexity value of 3, so the program is easy to read. The total time spent going through the entire scraping process amounts to around 10 minutes and 45 seconds, but the time variable is dependent on the HTML page request connection. The success rate of all Arabic letters is 99.95%, which is caused by the loss of the first three derivative words on the Alif letter because they went undetected by the program, and they ended up not being processed. However, the other derivative words on the Corpus Quran web were able to be detected and processed as intended. In conclusion, this research stands as a significant progress between the Software Engineering field and the Arabic linguistic field. By extracting derivative words people may gain easier access to Quranic knowledges and feel deeper appreciation for the complexity of the Quranic language. This research may serve as a tool for understanding sacred texts and it also may remind all parties involved of the importance of data practices.

REFERENCES

[1] N. H. A. Shukri, M. K. M. Nasir, and K. Abdul Razak, "Educational Strategies on Memorizing the Quran: A Review of Literature," International Journal of Academic Research in Progressive Education and Development, vol. 9, no. 2, Jul. 2020, doi: 10.6007/IJARPED/v9-i2/7649.

[2] N. Himawan, G. Wasis Wicaksono, and I. Nuryasin, "Ekstraksi Fi'il dan Isim Pada Kaidah Nahwu Shorof Berbasis Android," REPOSITOR, vol. 2, no. 5, pp. 619–626, 2020.

[3] A. N. Qowim, "Metode Pendidikan Islam Perspektif Al-Qur'an," IQ (Ilmu Al-qur'an): Jurnal Pendidikan Islam, vol. 3, no. 01, pp. 35–58, Jul. 2020, doi: 10.37542/iq.v3i01.53.

[4] M. Ikhwan, "Legitimasi Islam: Sebuah Pembacaan Teoritis Tentang Wahyu Alquran," MUTAWATIR, vol. 10, no. 1, pp. 144–169, Jun. 2020, doi: 10.15642/mutawatir.2020.10.1.144-169.

[5] "Arabic language | History & Alphabet | Britannica." Accessed: Dec. 09, 2023. [Online]. Available: https://www.britannica.com/topic/Arabic-language

[6] N. Mufidah, I. Izha, R. Pendidikan, B. Arab, U. M. Malik, and I. Malang, "PENGAJARAN KOSA KATA UNTUK MAHASISWA KELAS INTENSIF BAHASA ARAB (Vocabulary Teaching For Arabic Intensive Class)," 2020.

[7] E. Suhemi, "Mashdar dalam Surat Al-Kahfi: Suatu Kajian Morfologis," Jurnal Ilmiah Al-Mu'ashirah, vol. 17, p. 186, Oct. 2020, doi: 10.22373/jim.v17i2.9180.

[8] Kamalia, "PRONOMINA (ISIM DHAMIR) ATAU KATA GANTI DALAM BAHASA ARAB (TINJAUAN GENDER)," 2019. doi: 10.37064/ai.v7i2.7812.

[9] M. T. Ben Othman, M. A. Al-Hagery, and Y. M. El Hashemi, "Arabic Text Processing Model: Verbs Roots and Conjugation Automation," IEEE Access, vol. 8, pp. 103913–103923, 2020, doi: 10.1109/ACCESS.2020.2999259.

[10] V. Krotov, L. Johnson, and L. Silva, "Legality and Ethics of Web Scraping," Communications of the Association for Information Systems, vol. 47, pp. 539–563, 2020, doi: 10.17705/1CAIS.04724.

[11] M. Khder, "Web Scraping or Web Crawling: State of Art, Techniques, Approaches and Application," International Journal of Advances in Soft Computing and its Applications, vol. 13, no. 3, pp. 145–168, Dec. 2021, doi: 10.15849/IJASCA.211128.11.

[12] G. W. Noblit, D. Beach, B. Bueno, L. Fickel, W. Pillow, and M. Thapan, The Oxford Encyclopedia of Qualitative Research Methods in Education. 2020.

[13] Subhan Hi Ali Dodego, "Pentingnya Penguasaan Bahasa Arab Dalam Pembelajaran Pendidikan Agama Islam," PESHUM : Jurnal Pendidikan, Sosial dan Humaniora, vol. 1, no. 2, pp. 55–70, Feb. 2022, doi: 10.56799/peshum.v1i2.48.

[14] "The Quranic Arabic Corpus - Word by Word Grammar, Syntax and Morphology of the Holy Quran." Accessed: Dec. 09, 2023. [Online]. Available: https://corpus.quran.com/

[15] O. Oueslati, E. Cambria, M. Ben HajHmida, and H. Ounelli, "A review of sentiment analysis research in Arabic language," Future Generation Computer Systems, vol. 112, pp. 408–430, Nov. 2020, doi: 10.1016/j.future.2020.05.034.

[16] "What is Python? Executive Summary | Python.org." Accessed: Oct. 22, 2023. [Online]. Available: https://www.python.org/doc/essays/blurb/

[17] "Introduction to Python." Accessed: Dec. 09, 2023. [Online]. Available: https://www.w3schools.com/python/python_intro.asp

[18] M. D. Squire et al., "Cyclomatic Complexity and Basis Path Testing Study," 2020.

[19] S. Huntsman, "Path homology as a stronger analogue of cyclomatic complexity," Mar. 2020.

[20] "Why good metrics values do not equal good quality." Accessed: Dec. 08, 2023. [Online]. Available: https://www.codecentric.de/wissens-hub/blog/why-good-metrics-values-do-not-equal-good-quality