Implementation of Web Scraping on Job Vacancy Sites Using Regular Expression Method

1st Fitriono Arya Riski
Informatics Department
Telkom University
Bandung, Indonesia
faryariski@student.telkomuniversity.ac.id

2nd Nungki Selviandro
Software Engineering Department
Telkom University
Bandung, Indonesia
nselviandro@telkomuniversity.ac.id

3rd Monterico Adrian
Information Technology Department
Telkom University
Bandung, Indonesia
monterico@telkomuniversity.ac.id

2022 1st International Conference on Software Engineering and Information Technology (ICoSEIT) | 978-1-6654-7303-3/22/$31.00 ©2022 IEEE | DOI: 10.1109/ICOSEIT55604.2022.10029964

***Abstract*—Job vacancy information is now not only found in newspapers, brochures, or visiting companies but information can be found on the internet. On the internet many third-party websites that provide job vacancy information that can help job seekers get information, some third party websites sometimes do not have vacancy information on the same company as other websites, because the company that owns the vacancy does not provide their company's job vacancy information to all third-party websites, so job seekers visit many websites to meet each other to get the information they want, so job seekers visit many websites to find each other to get the information they want, which takes a long time. To simplify and shorten the time in searching for information, we can utilize a web that can display more than one third-party web so that job seekers do not need to open many third-party webs. The purpose of this research is to implement Web Scraping to retrieve data about job vacancy information with Regular Expression method and build a web-based information system to display data obtained from Web Scraping. The results of Web Scraping experiments with the Regular expression method on three third-party websites namely Jobstreet, Kalibrr, and Glints were able to retrieve data from the three websites. The resulting accuracy is very good, on Jobstreet and Kalibrr produces 100% accuracy in providing correct results while Glints produces 99.4% accuracy. The data collected from the web scraping process is stored in the database and then displayed on the developed web.**

***Keywords*—web scraping, regular expression, information system, job vacancy**

I. Introduction

In the current era of technological development, with the advancement and increasing need for information in society, especially with the rapid development of internet technology. The internet has become an important source of information for the community. The information is in the form of information on technology, social, education, job vacancies, and so on. In addition, with the existence of technologies such as smartphones, computers, and tablets, users can use internet services comfortably and can be used anywhere.

Information on job vacancies is now not only found in newspapers, brochures or visiting companies, but information can be found through the internet. On the internet, there are many third-party websites that provide job vacancy information from various companies. This is one of the most common ways for job seekers to do online. Third-party webs are very helpful in making it easier for job seekers to get job vacancy information. However, some third-party webs sometimes have different information because the company that owns the vacancy does not provide information on their company's job vacancies to all third-party webs, thus making job seekers visit one by one web to get the desired information, this certainly takes a long time.

Web Scraping technique is a process of retrieving a semi-structured document from the internet, generally in the form of web pages written in HTML or XHTML markup language, and analyzing the document to retrieve certain data from the page that will be used for other purposes [1].

Some related research on the implementation of Web Scraping techniques has been done before, among others: the implementation of Web Scraping to retrieve news article data on three websites, the method used in this research is Regular Expression to recognize the elements searched on the news website [2], the implementation of Web Scraping with HTML DOM method to download scientific article publication data from the Google Scholar based on id or class contained in the Google Scholar web source code [3], the implementation of Web Scraping for data retrieval on Marketplace sites [4]. There are several methods that can be used in the implementation of Web Scraping such as Regular Expression, HTML DOM, and XPath [5]. In this research, we use Web Scraping to collect data about job vacancy information including job name, company name, location, requirements, posting date, company logo, and URL address from third-party websites namely Kalibrr, Jobstreet, and Glints. The method used in Web Scraping in this research is Regular Expression which is used to recognize those seven elements. The data that has been successfully obtained is stored in the database and then displayed on a web-based information system that is created.

II. Literature Review

There is a literature review to study this research, namely Job Vacancy Web, Web Scraping Python, Django Framework, and Regular Expression. Each will be discussed in the following subsections.

Authorized licensed use limited to: Slovak University of Technology Trial User. Downloaded on November 05,2023 at 15:31:09 UTC from IEEE Xplore. Restrictions apply.

A. *Job Vacancy Web*

Job vacancy is an information about a job vacancies created by various companies that contains certain requirements [6]. The purpose of the job vacancy web is to facilitate job seekers in finding the desired job vacancy information.

B. *Web Scraping*

Web Scraping or also known as web extraction is a technique for extracting or obtaining data from the World Wide Web (WWW) and saving it to a system file or database to be used as data analysis [7]. The implementation of Web Scraping only focuses on how to obtain data through data retrieval and extraction with varying data sizes [4]. Web scraping has been used for different purposes such as conducting online price comparisons, monitoring weather data, integrating data from multiple sources, research, extracting deals and discounts, extracting job vacancy information, collecting government data, and market analysis [8][9].

C. *Python*

Python is one of the many programming languages that can perform the execution process on several instructions directly, which is interpretative and is the most commonly used programming language in supporting various data science tasks, including web scraping [10][11].

D. *Django Framework*

Django is a free and open python-based web framework that follows the model-template-views (MTV) architecture pattern maintained by the Django Software Foundation (DSF). The main purpose of using Django is to make it easy to build complex, database-driven websites [10]. The Python programming language in this framework benefits from all Python libraries and can assure very good readability [12].

E. *Regular Expression*

Regular Expression or commonly called Regex is a pattern that contains characters (such as letters and numbers) and meta characters (such as symbols). Regular Expression can be used to search, replace, or extract data with identifiable patterns, for example, dates, postal codes, HTML tags, and so on [13][14].

III. System Design

In this section, we discuss the system design of this research. The system design includes Mapping Job Vacancy Web Pages, Developing Web Scraping Source Code, Developing Django Web Framework, Save the Scraping Data to the Database and System Testing.

A. *Mapping Job Vacancy Web Pages*

Display the HTML structure of the web page and analyze each class in the HTML through inspect element. In general, the job vacancy web has two main pages, namely the index page and the detail page. The index page is a page that contains all published job vacancy items, and the detail page is the page addressed by the address of each item that contains all information about the vacancy item.

Table 1, Table 2, and Table 3 below are the results of analyzing the HTML structure of the three job vacancy websites. There are differences in the elements used to present the job name, company name, company logo, location, requirements, posting date, and link of jobs.

The results of the job vacancy web mapping analysis shown in Table 1, Table 2, and Table 3 become the target in retrieving web scraping data using the Regular expression method.

TABLE I
The Tags of Jobs Vacancy Elements From Jobstreet

Element	HTML Tag
Title	<h1 class=”sx2jih0 _18qlyvc0 _18qlyvch _1d0g9qk4 _18qlyvcp _18qlyvc1x”>Nama Pekerjaan</h1>
Company Name	<span class=”sx2jih0 zcydq84u _18qlyvc0 _18qlyvc1x _18qlyvc2 _1d0g9qk4 _18qlyvcb”>Nama Perusahaan</span>
Company Logo	<img class=”Xir05_0” src=”Tautan Logo Perusahaan” alt=”Nama Perusahaan”>
Location	<span class=”sx2jih0 zcydq84u _18qlyvc0 _18qlyvc1x _18qlyvc1 _18qlyvca”>Lokasi Perusahaan</span>
Requirement	<div class=”sx2jih0”> Persyaratan Pekerjaan Yang Dituju</div>
Job Posted	<span class=”sx2jih0 zcydq84u _18qlyvc0 _18qlyvc1x _18qlyvc1 _18qlyvca”>Tanggal Posting Pekerjaan</span>
Link	<a href=”Tautan Menuju Halaman Pekerjaan” class=”_1hr6tkx5 _1hr6tkx8 _1hr6tkxb _1hr6tkxc _1hr6tkxf sx2jih0 sx2jihf zcydq8h”> </a>

TABLE II
The Tags of Jobs Vacancy Elements From Kalibrr

Element	HTML Tag
Title	<h1 itemprop=”title” class=”k-text-title k-inline-flex k-items-center md:k-text-primary-head md:k-flex lg:k-mt-16”>Nama Pekerjaan</h1>
Company Name	<a href=”Tautan Perusahaan”><h2 class=”k-inline-block”>Nama Perusahaan</h2></a>
Company Logo	<img loading=”lazy” src=”Tautan Logo Perusahaan” alt=”Kalibrr” class=”k-block k-max-w-full k-max-h-full k-bg-white k-mx-auto”>
Location	<span itemscope=”” itemtype=”http://schema.org/ PostalAddress”>Lokasi Perusahaan</span>
Requirement	<div itemprop=”qualifications” class=”k-mb-4 css-q0v7oq”>Persyaratan Pekerjaan Yang Dituju</div>
Job Posted	<div class=”k-text-subdued k-text-caption md:k-text-right”>Tanggal Posting Pekerjaan</div>
Link	<a class=”k-text-primary-color” itemprop=”name” href=”Tautan Menuju Halaman Pekerjaan”></a>

TABLE III
The Tags of Jobs Vacancy Elements From Glints

Element	HTML Tag
Title	<h1 class=”TopFoldsc__JobOver ViewTitle-sc-kklg8i-3 fFAcsE”>Nama Pekerjaan</h1>
Company Name	<div class=”TopFoldsc__JobOver ViewCompanyName-sc-kklg8i-5 eLQvRY”>Nama Perusahaan</div>
Company Logo	<div class=”TopFoldsc__Company Logo-sc-kklg8i-1 bFKRUY”img alt=”Company Logo” sizes=”70px” src=”Tautan Logo Perusahaan”></div>
Location	<div class=”TopFoldsc__JobOver ViewCompanyLocation-sc-kklg8i-6 gLATOW”><span><a href=”Tautan Lokasi”>Lokasi Perusahaan </a></span></div>
Requirement	<div class=”JobDescriptionsc__ DescriptionContainer-sc-1jylha1-2 lcBfuk”>Persyaratan Pekerjaan Yang Dituju</div>
Job Posted	<span data-recent=”false” class=”TopFoldsc__UpdatedAt-sc-kklg8i-12 bYndtI”>Tanggal Posting Pekerjaan</span>
Link	<a target=”_blank” class=”Compact Opportunity-Cardsc__CardAnchorWrapper-sc-1y4v110-18 iOjUdU job-search-results_job-card_link” href=”Tautan Menuju Halaman Pekerjaan”></a>

Authorized licensed use limited to: Slovak University of Technology Trial User. Downloaded on November 05,2023 at 15:31:09 UTC from IEEE Xplore. Restrictions apply.

B. Developing Web Scraping Source Code

Creation of data scraping code using Python language designed to retrieve data based on class elements in the HTML structure of the job vacancy web. The scraping process combines the Beautifulsoup library with the Regular Expression method to recognize certain class elements in HTML tags. Beautiful Soup is a python library used to parse and extract data from the HTML response. The data generated in the web scraping process is stored in the database using the MySQL connector library.

The web scraping process is done every three days with periodic scheduling, so the program runs automatically after three days. In addition, data retrieval is only done on the latest job vacancy information on each job vacancy web.

The following scraping flow map explains how the workflow in the data retrieval system on the web lo job vacancies. The scraping flow map can be seen in Figure 1.

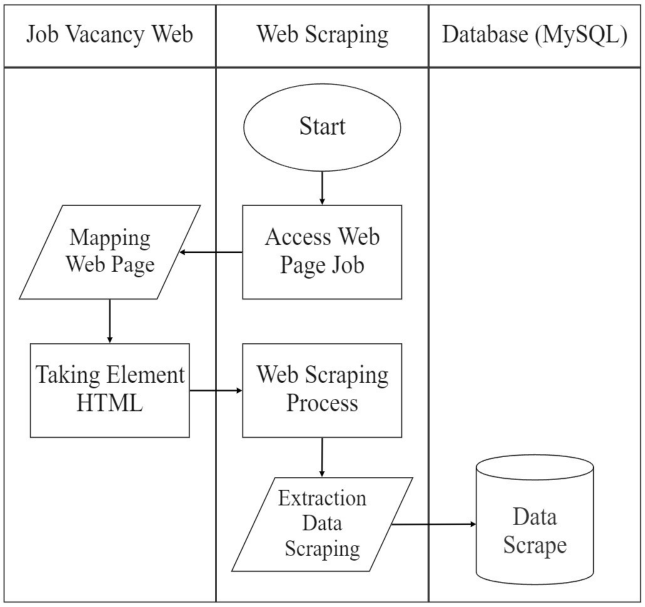


Fig. 1. Flow scraping.

C. Developing Django Web Framework

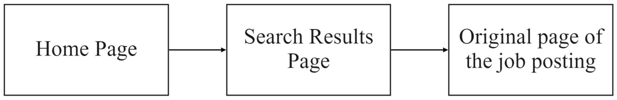


Fig. 2. Design architecture.

We use the Django framework to develop a website that is used to display information from scraping data stored in a database.

Regular Expression is not only used in web scraping code, to get comprehensive search results on the web when users are not sure of the right search keywords to use is an important module to develop a user-friendly web. The use of keyword-based search methodology alone is not enough to describe the information sought [10]. Developing web in this research, we use the proposed described in [10] by combining keyword-based search and Regular Expression to perform efficient search and comprehensive search results. The following page is displayed:

1) Home Page
   On the home page, there is a search field for the desired vacancy by entering the keyword.
2) Search Results Page
   A page that displays search results after entering the keyword you want to search for, the data displayed is the data stored in the database obtained in the scraping process. On this page, there is a detail button if you want to see information about the selected vacancy and an apply button to go to the original page of the vacancy.
3) Original page of the job posting
   Pages that lead to the original website of each targeted job opening.

D. Save the Scraping Data to the Database

Once the scraping process is complete, the data is saved into a MySQL database. The MySQL database is used as a tool to store data, as well as connect with the Django web.

E. System Testing

This research uses unit testing and black box testing. Unit testing is done to test the code on the web itself by writing test code.

Black box testing is done after unit testing to check and confirm whether the web functions as expected and to find errors that would be unexpected if done directly from the web User Interface (UI).

IV. Evaluation

In the evaluation section, we discuss the regular expression pattern used to retrieve data, the web developed for displaying the data scraping, the result of the experiment on the web scraping, and system testing with unit testing and black box testing..

A. Pattern of Regular Expression

After having tag elements for data retrieval based on analyzing the HTML structure of each job vacancy web, the next step is to build regular expression patterns to get job vacancy information. Based on Table I, Table II, and Table III, the Regular Expression patterns for the three job vacancy websites can be summarized in Table IV, Table V, Table VI.

Table IV until Table VI presents all Regular Expression patterns for job vacancy information extraction from the three job vacancy web pages. The Tables show that each job posting website used its own pattern for job name, company name, company logo, location, requirements, posting date, and link. Therefore, as seen in those tables, we create one Regular Expression pattern for each element for each job vacancy web.

An explanation of the Regular Expression pattern used in data retrieval. From the metacharacter Regular Expression is

Authorized licensed use limited to: Slovak University of Technology Trial User. Downloaded on November 05,2023 at 15:31:09 UTC from IEEE Xplore. Restrictions apply.

created in the following pattern (.*?). The pattern will get a matching on any string. This pattern is much easier and helpful when we have more complicated regular expressions.

TABLE IV
PATTERN OF REGULAR EXPRESSION FOR ELEMENT TAGS ON JOBSTREET

Element	Pattern
Job Name	<h1 class=”sx2jih0 _18qlyvc0 _18qlyvch _1d0g9qk4 _18qlyvcp _18qlyvc1x”>(.*?)</h1>
Company Name	<span class=”sx2jih0 zcydq84u _18qlyvc0 _18qlyvc1x _18qlyvc2 _1d0g9qk4 _18qlyvcb”>(.*?)</span>
Company Logo	<img class=”Xir05_0” src=”(.*?)” alt=”Nama Perusahaan”>
Location	<span class=”sx2jih0 zcydq84u _18qlyvc0 _18qlyvc1x _18qlyvc1 _18qlyvca”>(.*?)</span>
Requirement	<div class=”sx2jih0”> Persyaratan Pekerjaan Yang Dituju</div>
Posting Date	<span class=”sx2jih0 zcydq84u _18qlyvc0 _18qlyvc1x _18qlyvc1 _18qlyvca”>(.*?)</span>
Link	<a href=”(/id/job/.*?)” class=”_1hr6tkx5 _1hr6tkx8 _1hr6tkxb sx2jih0 sx2jihf zcydq8h”> </a>

TABLE V
PATTERN OF REGULAR EXPRESSION FOR ELEMENT TAGS ON KALIBRR

Elemen	Pattern
Job Name	<h1 itemprop=”title” class=”k-text-title k-inline-flex k-items-center md:k-text-primary-head md:k-flex lg:k-mt-16”>(.*?)</h1>
Company Name	<a href=”/id-ID/.*”><h2 class=”k-inline-block”>(.*?)</h2> </a>
Company Logo	<img loading=”lazy” src=”(https://.*?)” alt=”Kalibrr” class=”k-block.*”>
Location	<span itemscope=”” itemtype=”http://schema.org/PostalAddress” >(.*?)</span>
Requirement	<div itemprop=”qualifications” class=”k-mb-4.*”>Persyaratan Pekerjaan Yang Dituju</div>
Posting Date	<div class=”k-text-subdued k-text-caption md:k-text-right”>(.*?)</div>
Link	<a class=”k-text-primary-color” itemprop=”name” href=”(.*?)”></a>

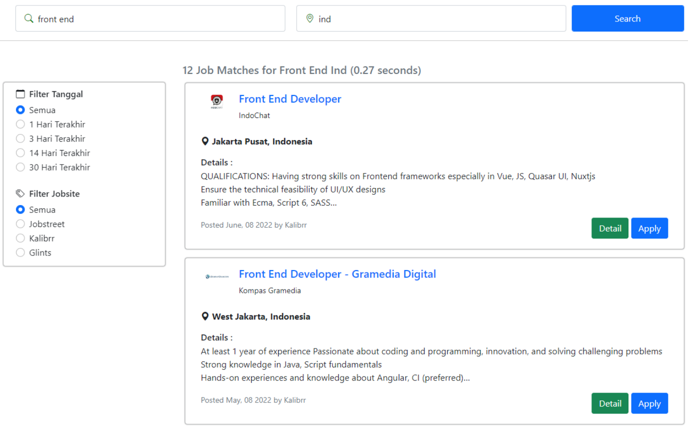
TABLE VI
PATTERN OF REGULAR EXPRESSION FOR ELEMENT TAGS ON GLINTS

Element	Pattern
Job Name	<h1 class=”TopFoldsc__JobOverViewTitle-sc-kklg8i-3 fFAcsE”>(.*?)</h1>
Company Name	<div class=”TopFoldsc__JobOverView CompanyName-sc-kklg8i-5eLQvRY”> (.*?)</div>
Company Logo	<div class=”TopFoldsc__CompanyLogo-sc-kklg8i-1 bFKRUY”>< img alt=”Company Logo” sizes=”70px” src=”(.*?)”></div>
Location	<div class=”TopFoldsc__JobOver ViewCompanyLocation-sc-kklg8i-6 gLATOW”> <span><a href=”/id/location/.*”>(.*?)</a></span> </div>
Requirement	<div class=”.*DescriptionContainer.*”> Persyaratan Pekerjaan Yang Dituju</div>
Posting Date	<span data-recent=”false” class=”TopFoldsc__UpdatedAt-sc-kklg8i-12 bYndtI”>(.*?)</span>
Link	<a target=”_blank” class=”CompactOpportunityCardsc__ CardAnchorWrapper-sc-1y4v110-18 iOjUdU job-search-results_job-card_link” href=”(.*?)”></a>

B. Developed Web Result

This web was developed to display job vacancy information from three job vacancy websites in Indonesia. Job seekers can find vacancy information easily without the need to open many job vacancies web because the web that was built has a feature to display based on three job vacancies and an apply button to visit the original page of the vacancy.

Figure 3 is a display of search results on the web. The search engine here combines the use of keyword-based with Regular Expression-based. From the Figure 3, the location search box is only the string {ind}. The string becomes a Regular Expression pattern to match and filter all locations that have available accommodations or matches. all locations that have accommodation or suitability available and the result found from the string is Indonesia.


front end
ind
Search

12 Job Matches for Front End Ind (0.27 seconds)

Filter Tanggal
● Semua
○ 1 Hari Terakhir
○ 3 Hari Terakhir
○ 14 Hari Terakhir
○ 30 Hari Terakhir

Filter Jobsite
● Semua
○ Jobstreet
○ Kalibrr
○ Glints

Front End Developer
IndoChat
Jakarta Pusat, Indonesia
Details :
QUALIFICATIONS: Having strong skills on Frontend frameworks especially in Vue, JS, Quasar UI, Nuxtjs
Ensure the technical feasibility of UI/UX designs
Familiar with Ecma, Script 6, SASS...
Posted June, 08 2022 by Kalibrr
Detail Apply

Front End Developer - Gramedia Digital
Kompas Gramedia
West Jakarta, Indonesia
Details :
At least 1 year of experience Passionate about coding and programming, innovation, and solving challenging problems
Strong knowledge in Java, Script fundamentals
Hands-on experiences and knowledge about Angular, CI (preferred)...
Posted May, 08 2022 by Kalibrr
Detail Apply

Fig. 3. Search results with Regular Expression.

C. Web Scraping Experiments

To experiment with web scraping, we need more than one web page per website. In this experiment, we tried to collect 10 pages and 10 different job categories for three job websites with 6 iterations. This experiment also compares with other methods such as XPATH, and CSS Selector. The experiment focused on the execution time of the web scraping process, memory usage and the amount of data obtained. Table VIII, Table IX, and Table X show the values generated by the three methods for comparison.

Table VII displays the results of the execution of web scraping on the Jobstreet web. From the test results obtained data that the Regular Expression method has an average time of 3.736 s, an average amount of data 2842, and average memory usage of 943.777.792 bytes. The memory usage of web scraping in experiments on the Jobstreet web is greater than on the Kalibrr web and the Glints web.

Table VIII displays the results of web scraping execution on the Kalibrr web. The results obtained in the test The Regular Expression method has an average time of 1.923 s, an average amount of data of 629, and average memory usage of 193.753.771 bytes.

Table IX displays the results of web scraping execution on the Glints web. The results of this test obtained data that the Regular Expression method has an average time of 12.177 s, an average amount of data of 2938, and an average memory usage of 583.248.555 bytes. The experiment web scraping on the Glints web takes quite a long time compared to the other

Authorized licensed use limited to: Slovak University of Technology Trial User. Downloaded on November 05,2023 at 15:31:09 UTC from IEEE Xplore. Restrictions apply.

TABLE VII
COMPARISON OF VALUE ON JOBSTREET

Experiment	REGEX			XPATH			CSS		
	Time (s)	Total Data	Memory (bytes)	Time (s)	Total Data	Memory (bytes)	Time (s)	Total Data	Memory (bytes)
1	3657.98	2835	951,672,832	4096.96	2844	2,634,608,640	3937.71	1977	933,711,872
2	3809,21	2844	943.783.936	4.375	2844	2.794.778.624	4050,02	1978	928.669.696
3	3537,11	2844	948.641.792	4010,54	2844	2.650.537.984	3987,06	1979	942.669.824
4	3669,91	2844	935.280.640	4277,48	2844	2.648.690.688	4046,4	1981	932.253.696
5	3794,47	2844	935.378.944	3919,04	2814	2.622.840.832	3861,63	1959	939.020.288
6	3950,05	2844	947.908.608	4084,07	2842	2.637.099.008	3904,14	1980	942.542.848
AVG	3.736	2842	943.777.792	4.127	2838	2.664.759.296	3.964	1975	936.478.037

TABLE VIII
COMPARISON OF VALUE ON KALIBRR

Experiment	REGEX			XPATH			CSS		
	Time (s)	Total Data	Memory (bytes)	Time (s)	Total Data	Memory (bytes)	Time (s)	Total Data	Memory (bytes)
1	1937,91	631	195.596.288	1927,45	631	338.432.000	1913,37	631	190.861.312
2	1925,01	631	196.247.552	1932,1	631	339.836.928	1917,36	631	193.556.480
3	1912,14	631	196.399.104	1929,57	631	337.817.600	1912,05	631	190.132.224
4	1931,91	631	194.240.512	1924,99	631	340.295.680	1935,19	631	189.599.744
5	1901,14	628	189.116.416	1957,21	623	339.546.112	1906,83	623	190.545.920
6	1929,84	623	190.922.752	1947,94	623	330.661.888	1920,39	623	191.143.936
AVG	1.923	629	193.753.771	1.937	628	337.765.035	1917,53	628	190.973.269

TABLE IX
COMPARISON OF VALUE ON GLINTS

Experiment	REGEX			XPATH			CSS		
	Time (s)	Total Data	Memory (bytes)	Time (s)	Total Data	Memory (bytes)	Time (s)	Total Data	Memory (bytes)
1	9497,77	2755	592.543.744	10703,13	2993	2.620.960.768	9799,8	2710	552.595.456
2	9264,9	2978	640.602.112	12765,58	2994	3.735.568.384	12595,77	2960	607.838.208
3	13634,97	2979	544.215.040	12984	2995	3.186.499.584	13669,94	2963	540.606.464
4	14696,94	2962	563.654.656	14998,67	2999	2.274.480.128	14594,91	2966	573.411.328
5	14588,01	2973	527.130.624	14853,46	2994	2.445.148.160	14312,82	2917	534.761.472
6	11380,77	2983	631.345.152	14683,49	2996	1.895.170.048	12285,44	2952	644.419.584
AVG	12.177	2938	583.248.555	13.498	2995	2.692.971.179	12876,44	2911	575.605.419

two webs. This is because, on the Glints web there are hidden elements, namely the date of posting vacancies, so the scraping program needs a combination with the selenium web driver library to perform implicit waits in identifying these hidden elements.

D. Unit Testing for Web

Table X shows Unit Test testing on components or units in a web-based information system. The tests carried out include testing the front page, testing the search results page, and testing the search data or keywords entered to search for information about job vacancies. The three tests carried out resulted in an "OK" status, which means that the test runs properly without any errors.

TABLE X
UNIT TESTING FOR WEB

Test	Status
Home Page Test Displayed	OK
Search Results Page Test Displayed	OK
Data Search Test	OK

E. Black Box Testing for Web

Table XI shows testing using Black Box. Testing is carried out on features on the web to check whether they are running correctly as expected or not.

TABLE XI
BLACK BOX TESTING FOR WEB

Test Case	Expected Results	Test Result
Search for job vacancies by entering keywords	The system will lead to the search results page according to the keyword	Success
Pressing the details button on the job vacancy card	The system will display a modal containing information about the job vacancy	Success
Pressing the apply button on the job vacancy card	The system will enter the original web page of the job vacancy	Success
Pressing one of the options in the last updated filter	The system will display data according to the selected last updated filter option	Success
Pressing one of the options in the Jobsite filter	The system will display data according to the selected Jobsite filter option	Success

Authorized licensed use limited to: Slovak University of Technology Trial User. Downloaded on November 05,2023 at 15:31:09 UTC from IEEE Xplore. Restrictions apply.

F. Analysis of Experimental Results

The results of the analysis of testing the implementation of Web Scraping on job vacancy sites that have been carried out are as follows.

Based on Tables VII, Table VIII, and Table IX, web scraping with the Regular Expression method is able to retrieve data on the targeted web. This experiment was conducted by comparing two other methods, namely Xpath and CSS Selector, from the data in the table it can be seen that each method produces different values. In the experiment, internet speed certainly affects the time in completing the web scraping execution process, so it is not only in terms of algorithms. The number of pages collected produces more data and the time required is quite long in the web scraping process.

In addition, we get the accuracy generated by Regular Expression in retrieving data. The accuracy calculation is done by comparing the total results obtained with the actual or expected total. Data retrieval on Jobstreet and Kalibrr produces 100% accuracy in providing correct results, while Glints data retrieval produces an accuracy of 99.4%. The accuracy results obtained from the method used provide good results in providing job vacancy information from three job vacancy websites.

V. Conclusion

The current web scraping program is still dependent on the targeted job vacancy web server. If the web server has problems, then the web scraping program also has problems in making requests. If there are changes in the HTML structure of the job vacancy web, the web scraping program also needs to change these elements to retrieve data and avoid errors during the web scraping process.

The web-based information system was developed to make it easier for job seekers to get job vacancy information without visiting job vacancy websites one by one so that it can shorten the time to find the desired vacancy information.

Based on the experimental results and analysis of the research conducted, the web scraping process has run well. Each job vacancy website has a different HTML structure for presenting job vacancy information. The implementation of web scraping with the Regular Expression method can be used on the three job vacancy websites to find related HTML elements on web pages and retrieve data. The experiments conducted on the three job vacancy websites resulted in different performances. However, data retrieval on Jobstreet and Kalibrr provide 100% correct data retrieval while Glints provides 99% correct results. Therefore, it is clear that the method used could satisfy in providing job vacancy information from HTML web pages without manually copying and pasting from the three job vacancy websites (Jobstreet, Kalibrr, and Glints). Thus, this method can be extended to other job websites and can be implemented in the real world.

References

[1] M. Turland. *Php | architect's Guide to Web scraping with PHP. Introduction-Web Scraping Defined, str, 2.* 2010.

[2] Achmad Maududie, Windi Eka Yulia Retnani, and Muhamat Abdul Rohim. An approach of web scraping on news website based on regular expression. In *2018 2nd East Indonesia Conference on Computer and Information Technology (EIConCIT)*, pages 203–207, 2018.

[3] Alam Rahmatulloh and Rohmat Gunawan. Web scraping with html dom method for data collection of scientific articles from google scholar. *Indonesian Journal of Information Systems*, 2(2):95–104, 2020.

[4] Dhita Deviacita A Yani, Helen Sasty Pratiwi, and Hafiz Muhardi. Implementasi web scraping untuk pengambilan data pada situs marketplace. *JUSTIN (Jurnal Sistem dan Teknologi Informasi)*, 7(4):257–262, 2019.

[5] Rohmat Gunawan, Alam Rahmatulloh, Irfan Darmawan, and Firman Firdaus. Comparison of web scraping techniques: regular expression, html dom and xpath. In *International Conference on Industrial Enterprise and System Engineering (IcoIESE 2018) Comparison*, volume 2, pages 283–287, 2019.

[6] Dwi Putra Githa Ketut Arif Suidiantara, I Nyoman Piarsa. Notification features on android-based job vacancy information system. *International Journal of Computer Applications Technology and Research(IJCATR)*, 8:429 – 434, 2019.

[7] Veronica Ambassador Flores, Putri Agung Permatasari, and Lie Jasa. Penerapan web scraping sebagai media pencarian dan menyimpan artikel ilmiah secara otomatis berdasarkan keyword. *Majalah Ilmiah Teknologi Elektro*, 19(2):157–162, 2020.

[8] De S Sirisuriya et al. A comparative study on web scraping. 2015.

[9] Erdinç Uzun. A novel web scraping approach using the additional information obtained from web pages. *IEEE Access*, 8:61726–61740, 2020.

[10] Ikechukwu Onyenwe, Stanley Ogbonna, Ebele Onyedimma, Onyedikachukwu Ikechukwu-Onyenwe, and Chidinma Nwafor. Developing smart web-search using regex. *arXiv preprint arXiv:2110.04767*, 2021.

[11] Sebastian Raschka, Joshua Patterson, and Corey Nolet. Machine learning in python: Main developments and technology trends in data science, machine learning, and artificial intelligence. *Information*, 11(4):193, 2020.

[12] Samuel Dauzon, Aidas Bendoraitis, and Arun Ravindran. *Django: web development with Python*. Packt Publishing Ltd, 2016.

[13] Gayathri Rajagopalan. Regular expressions and math with python. In *A Python Data Analyst's Toolkit*, pages 77–99. Springer, 2021.

[14] Erdinç Uzun. A regular expression generator based on css selectors for efficient extraction from html pages. *Turkish Journal of Electrical Engineering and Computer Sciences*, 28(6):3389–3401, 2020.

Authorized licensed use limited to: Slovak University of Technology Trial User. Downloaded on November 05,2023 at 15:31:09 UTC from IEEE Xplore. Restrictions apply.