Atlantis Highlights in Engineering (AHE), volume 2
International Conference on Industrial Enterprise and System Engineering (IcoIESE 2018)

Comparison of Web Scraping Techniques: Regular Expression, HTML DOM and Xpath

Rohmat Gunawan
Department of Informatics
Siliwangi University
Tasikmalaya, Indonesia
rohmatgunawan@unsil.ac.id

Alam Rahmatulloh
Department of Informatics
Siliwangi University
Tasikmalaya, Indonesia
alam@unsil.ac.id

Irfan Darmawan
Department of Information System
Telkom University
Bandung, Indonesia
irfandarmawan@telkomuniversity.ac.id

Firman Firdaus
Department of Informatics
Siliwangi University
Tasikmalaya, Indonesia
varminz@gmail.com

***Abstract*—Data collection is the initial stage of research. There are various data sources on the internet that can be used in the research process. The process of taking data or information from sites on the internet is called web scraping. Some methods of web scraping include Regular Expression (Regex), HTML DOM and XPath. This study aims to determine the performance of the three methods of web scraping. The Comparison is done by testing each method when retrieving data from the target website, then measuring the performance of the process and comparing it. Process time, memory usage, and data consumption are used as measurement parameters in the experiment. The results of the experiment show that web scraping with the regex method is the smallest in memory usage compared to the HTML DOM method, and Xpath. While HTML DOM requires the least amount of time and the smallest data consumption compared to Regular Expression and XPath methods.**

Keywords:DOM, Regex, Web Scraping, Xpath

I. INTRODUCTION

In the business, marketing, engineering, social sciences, or other fields of study, data plays an important role, which can be used as a basic reference in all processes involving the use of information and knowledge. Data collection is the initial stage of research, then measurement of information about interesting variables, in a systematic mode that allows someone to answer questions, express research questions, test hypotheses, and evaluate results [1]. Depending on the discipline or field of science, the nature of the information sought, and the goals or objectives of the user, data collection methods will vary. The approach to applying the method can also vary, adjusted for applicable objectives and circumstances, without sacrificing data integrity, accuracy and reliability.

There are various data sources on the internet that can be used in the research process. The process of taking data or information from sites on the internet is called web scraping [2],[3], [4], [5], [6], [7], web extraction [8],[9], [10],[11], web harvesting [12], [13]. Web scraping has been used widely and for different purposes including online price comparison, weather data monitoring, website change detection, research, integrating data from multiple sources, extract offers and discounts, scrape job postings information from job portals, brand monitoring, collect government data and market analysis [14].

Various web scraping methods have been developed in various studies, including: traditional copy and paste[14], Regular Expression (Regex)[14], Hypertext Markup Language Document Object Model (HTML DOM)[10], [14], [15] and XPath [4], [9]. The copy-pasting method is easy to do by opening the website in the browser, then copy and paste it on other media manually. This method is very simple and not difficult, but it cannot be done if the website has a barrier program[14], time selection of objects or texts that are relatively long, and done manually. While the Regex method, HTML DOM, XPath is more complicated and requires additional program before it can be used.

Development of web scraping methods has been carried out in various studies, but the performance of these methods is not yet known when the data scraping process is one of the interesting things to study. In this study, the web scraping of the Regex method, HTML DOM and XPath will be carried out by using time, memory usage and data usage parameters. The data that is sampled in this research is taken from one of the special webs that provides data services for the scraping process, namely `http://testing-ground.scraping.pro`.

II. WEB SCRAPING METHOD

A. Regular Expression (Regex)

Regular Expression (Regex) is a formula with a specific pattern that describes a set of words above several alphabets [16]. Regex can be used to match certain character patterns in a set of strings [16]. There are two types of regular expressions namely ordinary characters and metacharacters.

B. HTML DOM

Hyper Text Markup Language Document Object Model (HTML DOM) is a standard for getting, changing, adding, or deleting HTML elements[17]. DOM performance is by defining objects and properties of all HTML elements, with methods to access them. With DOM, JavaScript can access all elements in an HTML document. HTML DOM uses programming languages to access objects, usually JavaScript. All HTML elements are treated as objects. The programming interface is the method and property of each object.

C. XPath

XPath is the main element in the XSLT standard (Stylesheet Language Transformation). XPath can be used to navigate elements and attributes in eXtensible Markup

Language (XML) documents [18]. XPath is a language for selecting nodes in XML documents, can also be used with HTML. The most useful XPath expression is the location path. A path location at least uses one step location to identify a set of nodes in the document. The simplest location path is one that selects the document root node. This road is just a slash "/". The symbol is the root of the Unix system file and also the root node of a document.

III. METHODLOGY

There are 8 steps taken in this study, as shown in figure 1.

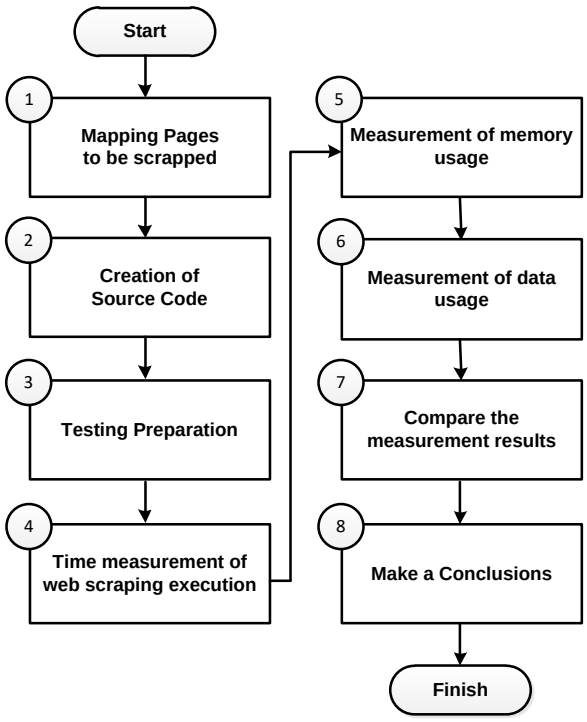

Figure 1. Stages of Comparative Research On Web Scraping Methods

A. *Mapping Pages to be Scrapped*

The mapping of source web pages to be captured data is done by displaying the source code of web pages through a web browser. Then identify all the id on the page element. The result identification id will be used to run the HTML DOM and XPath methods. Figure 2 shows an example of some of the id elements identified on the target web site.

```
1  </script>
2      <!--[if lt IE 7]>
3          <p class="chromeframe">You are usi
4      <![endif]-->
5      <div id="topbar"></div>
6      <a href="http://testing-ground.scrapir
7          <div id="title">WEB SCRAPER TESTIN
8          <div id="logo"></div>
9      </a>
10     <div id="content"><div class="caseblo
```

Figure 2. Source Code Of The Target Site's Web Page Marked On The ID Element

B. *Creation of Source Code*

In this study, making code is done using the Java programming language with the Standard Edition version. Some Java libraries are selected to process HTML requests, parse text, and make measurements. Pseudo code for each web scraping method used in the experiment is shown in figure 3-5.

```
String url = http://testing-
ground.scraping.pro;
String response = request html from url;
Arraydatarow;
Array datalist;
String[] parseresult = Pattern check
"<tr>(.*?)</tr>"on response;
for countfromparseresult do
  datarow[] = parseresult;
endfor
for countfromdatarow do
   String[] parseresult1 = Pattern check
"<td>(.*?)</td>" ondatarow;
   For eachparseresult1 do
     Datalist[count fromparseresult1] =
parseresult1;
   endfor
endfor
```

Figure 3. Pseudo Code of Regex

```
String[] datarow =new String[72];
Document doc = request from ("http://testing-
ground.scraping.pro/table?products=10&years=10
&quarters=4");
Element content = doc element which has a
case_table id;
Elements tbody = content elements that have tr
tags;
Integer x=0;
For each element tbody do
   element i = tbody elements that have td;
   for each element i do
      datalist[numberfrom i] = contentfrom
elemen i;
   endfor
endfor
```

Figure 4. Pseudo code of HTML DOM

```
String [][] datalist1 = new String[6][22];
String url = "http://testing-
ground.scraping.pro/table?products=10&years=10
&quarters=4";
HtmlPage page = request html page from url;
Integer i = count tr,j count td;
For i do
   For j do
     Datalist [i][j] = take data from address
xpath"//*[@id='case_table']/table/tbody/tr[i]/
td[j]";
   Endfor
Endfor
```

Figure 5. Pseudo code of Xpath

C. *Testing Preparation*

Preparations made at this stage include: java based application preparation that contains three methods to be tested that have been installed on a PC or laptop, internet connection and target web site for scraping: `http://testingground.scraping.pro`.

D. Time measurement of Web Scraping Execution

The time measurement is done by initializing the t0 variable before the code execution and initializing t1 after the execution of the method code and then doing the reduction operation (t1-t0). Pseudo code for time measurement is shown in figure 6.

```
Long t0=System.currentTimeMillis();
WebScrapingMethod();
Long t1=System.currentTimeMillis();
Return t1-t0;
```

Figure 6. Pseudo Code of Time Measurement Of WebScraping Execution.

E. Measurement of Memory Usage

Measurement of memory usage is done by initializing variable m0 before execution method code and initialization m1 after execution of method code, then search (m1-m0). Pseudo code for memory measurement shown in figure 7.

```
m0=Runtime.getRuntime().totalMemory()-
Runtime.getRuntime().freeMemory();
WebScrapingMethod();
m1=Runtime.getRuntime().totalMemory()-
Runtime.getRuntime().freeMemory();
return m1-m0;
```

Figure 7. PseudoCode of Memory Usage Measurement

F. Measurement of Data Usage

Measurement of data usage is done by using jnet library jnetpcap, which is a library to do packet sniffing through the network. The jnetpcap java library's source code is inserted before the source code of the method is performed, after the method completes, the sniffing process is stopped, and in large packets, the data packets are obtained, such as showed in Figure 8 and detail sniffing showed in figure 9.

```
starCapture();
WebScrapingMethod();
thread.stop();
```

Figure 8. PseudoCode of Measurement Data Usage

```
void starCapture(){
thread = new Thread(){
  public void run(){
   Pcap.findAllDevs(alldevs, errbuf);
   PcapIf device = alldevs.get(1);
   Pcap pcap = Pcap.openLive(device.getName(),
(64*1024), Pcap.MODE_PROMISCUOUS, (10*1000),
errbuf);
 pcap.loop(-1,jpacketHandler," ");
 }
 };
thread.start();
}
```

Figure 9. Pseudo Code Sniffing Method to Measure Data Usage

G. Compare The Measurement Results

The measurement results of each experiment, collected and taken the average value of each parameter. Thenconducted a comparison of experimental data between the three methods used.

H. Make a Conclusion

Analyze the test data between the three methods used then determine which is better for each parameter tested.

IV. Result And Analysis

In this section presented data of experimental results that have been done. Each method is chosen for speculative execution of web scraping; then the results are recorded for each of the predefined parameters.

A. Time Measurement

Table I displays the measurement data of the web scraping execution time for each method. The final row of the table shows the average time of execution after 20 test. From the experimental results obtained data as follows: regex method has an average time of 399.75 ms or 0.39 seconds, the DOM HTML method has an average time of 298.55 ms or 0.29 seconds, and XPath method has an average time 435.15 ms or 0.43 seconds.

Table I. Execution Time Measurement Result

Experiment	Time (ms)		
	REGEX	HTML DOM	XPATH
1	375	297	406
2	375	250	407
3	390	360	485
4	391	281	859
5	391	454	422
6	382	265	390
7	446	266	406
8	322	281	422
9	325	265	406
10	377	500	438
11	294	250	391
12	286	265	375
13	422	266	640
14	390	266	390
15	516	250	375
16	406	266	375
17	406	250	375
18	594	282	375
19	391	407	391
20	516	250	375
Avg	**399,75**	**298,55**	**435,15**

Table I displays the measurement data of the web scraping execution time for each method. The final row of the table shows the average time of execution after 20 test. From the experimental results obtained data as follows: regex method has an average time of 399.75 ms or 0.39 seconds, the DOM HTML method has an average time of 298.55 ms or 0.29 seconds, and XPath method has an average time 435.15 ms or 0.43 seconds.

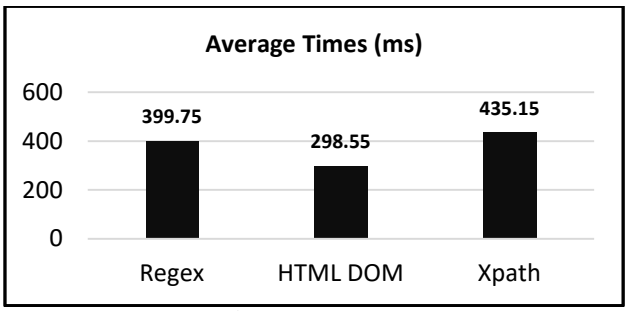

Figure 10. Average Time of Measurement Results

Figure 10 shows the results of calculating the average execution time. It is known that the HTML DOM method requires the least amount of time compared to the Regex or Xpath method.

B. The Measurement Result of Memory Usage

Table II displays the data, the use of memory at the time of execution or scraping the web for each method. Fromthe experimental results obtained data as follows: regex method average use memory of 564 782,5 bytes or 564KB; the average HTML DOM method uses the memory of 4,817,132 bytes or 4.8 MB; the average XPath method uses 574,546.4 bytes or 574 KB of memory.

Table II. Measurement Results of Memory Usage

Experiment	Memory Usage (bytes)		
	REGEX	HTML DOM	XPATH
1	513.264	4.699.048	713.320
2	664.416	4.739.912	505.904
3	625.552	4.614.448	625.584
4	461.368	5.084.552	486.008
5	572.576	4.707.232	923.528
6	592.123	4.618.816	477.800
7	722.345	4.743.344	461.400
8	772.364	4.703.696	584.720
9	547.326	4.743.400	694.072
10	682.483	4.730.112	461.400
11	469.576	5.098.016	656.416
12	485.976	4.684.760	584.232
13	505.176	5.090.696	469.608
14	489.472	5.091.936	584.752
15	485.976	5.087.408	462.576
16	461.416	4.652.704	461.400
17	461.368	4.679.448	625.672
18	461.368	5.084.200	625.584
19	664.496	4.836.776	625.552
20	657.008	4.652.144	461.400
Avg	**564.782,5**	**4.817.132**	**574.546,4**

Figure 11 shows the results of calculating the average memory usage. It is known that the Regex Method requires the least memory compared to the HTML DOM or XPath method.

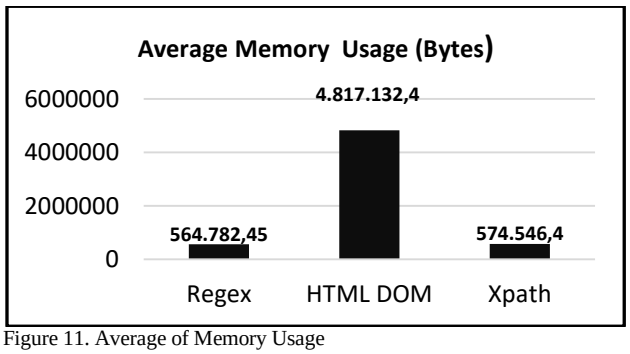

Figure 11. Average of Memory Usage

C. The Results of Data Usage Measurement

Table III displays the data, the use of data at the time of execution or web scraping for each method. From theexperimental data results as follows: regex method average using data amounted to 50.295,05 bytes or 50,29 KB; theaverage HTML DOM method uses data of 8,803.3 or 8.9 KB; the XPath method uses data of 17,769.85 bytes or 17.7 KB.

Table III. Measurement Results of Data Usage

Experiment	Data Usage (Bytes)		
	REGEX	HTMLDOM	XPATH
1	22.155	6.285	24.790
2	26.758	6.285	11.037
3	27.629	6.671	10.293
4	31.596	5.911	9.705
5	26.758	6.993	19.005
6	53.245	6.297	22.165
7	34.165	6.297	25.921
8	22.456	6.619	23.324
9	34.274	10.529	13.236
10	124.341	8.539	10.039
11	307.084	16.614	9.487
12	52.223	11.749	19.649
13	29.628	16.355	23.625
14	30.188	18.747	18.543
15	32.796	7.257	13.725
16	32.184	6.671	10.963
17	32.875	8.240	11.547
18	28.872	5.911	20.987
19	28.180	7.489	27.246
20	28.494	6.607	30.110
Avg	**50.295,05**	**8.803,3**	**17.769,85**

Figure 11 shows the results of calculating the average data usage. It is known that the HTML DOM requires the least memory compared to the Regex or Xpath.

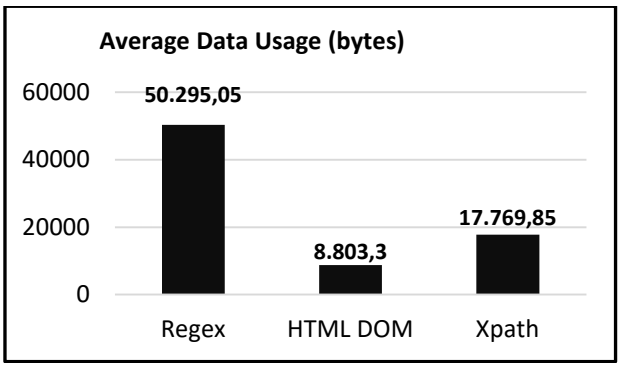

Figure 11. Average Data Usage

After the experiment for each method selected and the average value calculated for each parameter, then it is compared to find out the performance based on the three parameters selected, as shown in table 4.

Table IV. Comparison of The Average Value Of Each Parameter

Parameter	REGEX	HTML DOM	XPATH
Time (Avg)	399,75	**298,55**	435,15
MemoryUsage (Avg)	**564.782,5**	4.817.132	574.546,4
DataUsage (Avg)	50.295,05	**8.803,3**	17.769,85

From the data in Table IV it can be seen that the regular expression method is the smallest in memory usage compared to the HTML DOM method, and XPath. While HTML DOM takes the least amount of time and uses the smallest data compared to Regex and XPath methods.

V. CONCLUSION

Based on the results of experiments in this study there are two main things obtained:

1. These three methods: regex, HTML DOM, XPath can be used to process web scraping, by searching for related HTML elements from the target web page.
2. The regular expression method is the smallest in memory usage compared to HTML DOM, and XPath methods. While HTML DOM takes the least time and uses the smallest data compared to regex and XPath methods.

VI. FUTURE WORK

Future challenges that can be done include comparing the performance of other web scraping methods, such as CSS selector, Vertical aggregation, Semantic Annotation Recognizing, Computer Vision web-page Analysis. The addition of other parameters in testing, repairing or combining methods to correct deficiencies of existing methods can be done to optimize the previous method.

REFERENCES

[1] Anastasia, "Overview of Qualitative And Quantitative Data Collection Methods," *Cleverism*, pp. 1–17, 2017.
[2] G. Gupta and I. Chhabra, "Optimized Template Detection and Extraction Algorithm for Web Scraping of Dynamic Web Pages," vol. 13, no. 2, pp. 719–732, 2017.
[3] S. Khalil and M. Fakir, "SoftwareX RCrawler : An R package for parallel web crawling and scraping," *SoftwareX*, vol. 6, pp. 98–106, 2017.
[4] G. Grasso, T. Furche, and C. Schallhart, "Effective Web Scraping with OXPath," pp. 23–25.
[5] E. Vargiu and M. Urru, "Exploiting web scraping in a collaborative filtering- based approach to web advertising," vol. 2, no. 1, pp. 44–54, 2013.
[6] R. S. Chaulagain, S. Pandey, S. R. Basnet, and S. Shakya, "Cloud Based Web Scraping for Big Data Applications," 2017.
[7] P. Meschenmoser, N. Meuschke, M. Hotz, and B. Gipp, "Scraping Scientific Web Repositories : Challenges and Solutions for Automated Content Extraction". September, pp. 1–15, 2017.
[8] E. Ferrara, P. De Meo, G. Fiumara, and R. Baumgartner, "Knowledge-Based Syste Web data extraction , applications and techniques : A survey," *Knowledge-Based Syst.*, vol. 70, pp. 301–323, 2014.
[9] T. Furche, G. Gottlob, G. Grasso, C. Schallhart, A. Sellers, and C. Foy, "OXPath : A Language for Scalable , Memory-efficient Data Extraction from Web Applications Scenario : History Books on Seattle," no. 1016, pp. 1016–1027, 2011.
[10] E. Uzun, T. Yerlİkaya, and O. Kirat, "Comparison Of Python Libraries Used For Web Data Extraction," no. May, 2018.
[11] P. Yesuraju *et al.*, "A Language Independent Web Data Extraction Using," pp. 635–639, 2013.
[12] Z. Li, X. Zhang, H. Huang, Q. Xie, J. Zhu, and X. Zhou, "Addressing Instance Ambiguity in Web Harvesting," *Proc. 18th Int. Work. Web Databases - WebDB'15*, pp. 6–12, 2010.
[13] N. Tandon, G. de Melo, F. Suchanek, and G. Weikum, "WebChild : Harvesting and Organizing Commonsense Knowledge from the Web," *Proc. 7th ACM Int. Conf. Web Search Data Min. (WSDM 2014)*, pp. 523–532, 2014.
[14] S. C. M. de S Sirisuriya, "A Comparative Study on Web Scraping," *Proc. 8th Int. Res. Conf. KDU*, no. November, pp. 135–140, 2015.
[15] M. K. Sarma, "A DOM-Tree based Representation of Web Document Structure for Web Mining Applications," no. July, pp. 1437–1439, 2002.
[16] A. Backurs and P. Indyk, "Which Regular Expression Patterns Are Hard to Match?," *Proc. - Annu. IEEE Symp. Found. Comput. Sci. FOCS*, vol. 2016–December, pp. 457–466, 2016.
[17] W3C, "What is the Document Object Model?," 2016. [Online]. Available: https://www.w3.org/TR/WD-DOM/introduction.html.
[18] X. P. Expressions, "XML and XPath," 2018. [Online]. Available: https://www.w3schools.com/xml/xml_xpath.asp.