TELKOMNIKA, Vol. 17, No. 1, February 2019, pp. 275 ~ 281
ISSN: 1693-6930, accredited First Grade by Kemenristekdikti, Decree No: 21/E/KPT/2018
DOI: 10.12928/TELKOMNIKA.v17i1.11613

Implementation of web scraping on GitHub task monitoring system

Rolly Maulana Awangga[*1], Syafrial Fachri Pane[2], and Restiyana Dwi Astuti[3]
Applied Bachelor Program of Informatics Engineering, Politeknik Pos Indonesia, Indonesia
[*]Corresponding author, e-mail: awangga@poltekpos.ac.id

Abstract

Evolution of information and technology increasingly sophisticated, also influential in the field of education. One of the implementation of information and technology development in the field of education is e-learning or electronic learning. GitHub social network can be one of the e-learning media in studying software development because GitHub provides access control The number of contributors who commits or change in a repository to make the duration of the calculation process to fill the parameter value that has been determined. Based on the issue, this research aims to build a page capable of integrating information from the GitHub repository page. Integration of information will be made by utilizing web scraping technology. With a web page that integrates information from the GitHub repository page to get repository, collaborators, commits, and issues information, the lecturer does not need to calculate how often the participant contributes to the task.

Keyword: *task management, web scraping, GitHub, python, e-learning.*

1. Introduction

Learning paradigm applies traditional system where a teacher is source of everything and it calls Teacher-centered and nowadays it has been changed become learning paradigm of Student-Centered where students are required to be active to elaborate the information that have been obtained previously and also sharpen their collaboration skills in solving problem creatively and skillfully [1]. Along with the statements, students must be able to do learning independently, so in learning activity needs to choose an appropriate strategy in order that learning process runs effectively and efficiently. An independent learning process can give benefit to improve student independence in order not depend on the attendance and the description of teaching materials from their teacher. The role of teacher in Student-Centered learning paradigm is as a facilitator who designs the learning process. Therefore, to support the learning activities with Student-Centered paradigm requires a media that can help and facilitate in doing collaboration [2] to generate stimulus for learners to expand, deepen, and apply the information that have been received while at the class [3].

Along with the development of Information and Communication that increase rapidly, it will also give impact to Educational Technology. With the existence of application technology in education field will indirectly also affect to the method of learning activities that are expected to help students. One of the integration products of information technology into the world of education is e-learning [4].

Implementation of e-learning as a medium of communication and recently learning has been developing in educational institutions, especially in college level [4]. Generally, universities that apply e-learning systems use it as supplement (addition) through subject material that presents regularly in the classroom. GitHub is one of (or can be) e-learning media in software development because GitHub provides control access, source control, collaborations, and transparency features such as bug tracking, feature request, task management, and issues [5]. Teacher take advantage of collaboration and transparency features from GitHub to create, reuse, and combine lessons to encourage contribution from students and monitor their activities on given

Received July 30, 2018; Revised October 10, 2018; Accepted November 13, 2018

tasks [6]. The level of student activeness towards their contribution to a project can be seen from regular commitment of students and the change on a repository. The implementation of web scraping on monitoring task system integrated with GitHub can help students and teachers to get information on repositories, collaborators, commits, and issues which can be shorten in the process of calculating the level of students' activeness according to their contribution that have been made within a certain period.

2. Related Works

Learning method based on Student-Centered Learning combines collaborative method by using learning media such as audio tapes, Overhead Transparency (OHT), or GitHub cited by Joseph Feliciano, Margaret-Anne Storey, Yiyun Zhao, Weiliang Wang and Alexey Zagalasky, in their research, they describes how GitHub emerges as a collaboration platform for education and it aims to understand how does the environment of GitHub supports social and collaborative features can fix (or may inhibit) experiences from student and teacher [7]. From the finding of their research, they find that students get benefit from transparent feature and open workflow from GitHub. However, some students worry because GitHub is not inherent learning media [6].

The implementation of web scraping as a data retrieval technique that has been done by previous research for various needs and objects, Leo Rizky Julian and Friska Natalia uses web scraping technique to compare data from five different online stores in computer, then the user can save the cost of purchasing components of computer [8]. The comparison features based on the principle of consumers who want to buy goods not only at the lowest price but also the best quality. Other research are conducted by K. Sundaramoorthy, R. Durga and S. Nagadarshini from Agni College of Technology, background of the research is simplify to categorize news from various portals, a bot is used dynamically for extracting URL at the specific interval [9].

After all explained related work, this research combines some of the above mentioned research, implementing web scraping with e-learning through GitHub. Web scraping will work to harvesting selected information that will become the criterion of a task assessment, and this method make it easier to check the task.

3. Methodology

3.1. Web Scraping

Web scraping, usually called web crawling or web spidering or programmatically going over a collection of web pages and extracting data [10] and also this method is excellent at gathering or collecting and processing large amounts of data [11]. This is a method employed to extract very large amounts of data from websites whereby the data is extracted and saved to a local file in your computer or to a database in table format or to a spreadsheet file. Web scraping services is the technique of automating this process, instead of manually copying the data from GitHub or any other website.

The layout of the GitHub is described using Hypertext Markup Language or HTML. HTML document mostly dwell of four type of elements; structure of document, inline, reciprocal elements, and block. The most common abstract model for HTML documents are trees, and the example of a HTML modelled as a tree shown in Figure 1.

3.1.1. Web Scraping Steps

To grab a data from each link using BeautifulSoup4 module on Python 3 and it needs a few stage. How much stage needed depends on a link or web structure. The first step is to determine pages which will be used as information sources. List of web page address that will be used in this study are shown by Table 1.

The second step is to extract the information from the source page by using web scraping technique. Generally, there are two stages to take data automatically from a webpage are as follows:

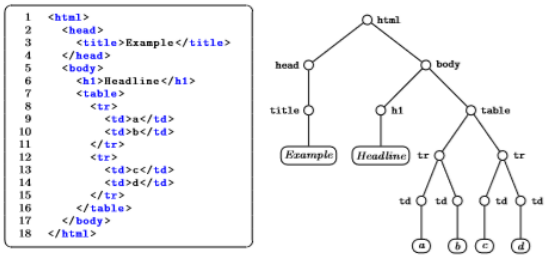

Figure 1. An Example of a HTML Document Modelled as a Tree

Table 1. The List of URL

URL	Spesific URL
https://github.com/	https://github.com/bukuinformatika/sig https://github.com/bukuinformatika/sig/issues https://github.com/bukuinformatika/sig/commits?author=awangga

1. Learning and identifying the HTML document from the information website that will be taken. HTML that flank the information that will be taken

2. Searching navigation mechanisms on the website to be retrieved for information to imitate through the web scraper application that will be created. In this stage, BeautifulSoup4 will extract multiple types of data - text, links and more as shown at Figure 2.

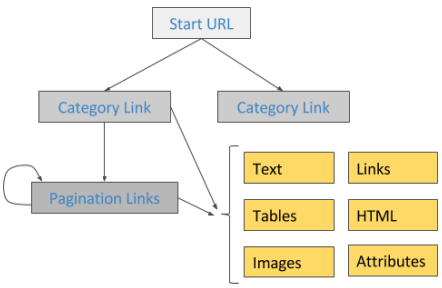

Figure 2. Graph of Web Scraping

3.1.2. Data Extraction

This step is the act of development of to retrieving data that is already exists on a website and convert it into a format that's suitable for analysis. Web-Pages are rendered by the browser. BeautifulSoup4 is essentially a set of wrapper functions that make it simple to select common HTML or XML elements.

3.1.3. Transformation

After author get the cleaned data from the parsing and cleaning, the data serialization module is used to serialize the data according to the data. This is the final module and result that will be transformed into a spreadsheet document and the lecturer will use the data for student task assessment.

4. Result and Analysis

4.1. Requirement Analysis

The first phase in the waterfall model is requirement analysis. In this phase, the author analyzes the requirements that will be made for the application and determines the features of the application. The requirement analysis phase is using three methods to obtain information about how should the application be made. The methods is observations and references studies.

4.2. Design

In the design phase, unified modeling language (UML) diagrams such as use case diagram, activity diagram, and sequence diagram were designed. This activity is to illustrate the process that runs in the application and the relations between entities in the application, and for the front-end, the author used Flask micro web framework written in Python and based on the Werkzeug toolkit and Jinja2 template [12].

4.2.1. The XML DOM TREE Generation

XML parsing is taking in XML code andextracting relevant information [13] like the title of the repository, fork count, issue count, contribution count, commit link, author name, date, and commits count XML parsing is the process of taking raw XML code, reading it, and generating a DOM tree object structure from it. BeautifulSoup4 is a set of wrapper functions that is used to select XML and HTML elements. It is a class that is used to parse the XML files directly also a DOM based toola in which the parser makes a single sequential pass through the file to parse the XML file [14]. The parser does not save any of the tags or the contents inside the tags. So it leads to super fast parsing because the XML file contents is not changed by the parser and the parser makes only one pass through the file.

BeautifulSoup4 class constructs a DOM (Document Object Model) object. It means that the entire contents of the XML file are stored in memory. DOM is a convention used in HTML, XHTML, and XML for representing and interacting with objects [14]. The elements in an XML document may have attributes. Even though it is a slower form of parsing, it allows making changes to XML file contents. BeautifulSoup4 uses two kinds of objects to perform XML parsing. The objects are BeautifulSoup4 and tag in order to do XML parsing using BeautifulSou4. BeautifulSoup4 is an object that holds the entire XML file's content in a tree-like structure, The tag object contains number of attributes and methods that manipulates the XML file easily.

4.3. Algorithm for Scraping the Content from XML Using BeautifulSoup4

1. Import the necessary libraries for scraping such as BeautifulSoup4 to parse the data returned from the website.
2. Fetch the links of the url using urllib2 library and save it in a variable.
3. For each link do:
   (a) Parse the XML in the page variable and store it in a BeautifulSoup4 format.
   (b) For each data in the item tag do scrap the title of the repository, fork count, issue count, contribution count, commit link, author name, date, and commits count.
4. Save the scraped content into the Microsoft Excel or spreadsheet format.

Listing 1. Scraping Commit URL

```
def parseCommitData():
global commit_url, commit_data
resp = ses.get(commit_url)
soup = BeautifulSoup(resp.text, "html.parser")
div_ele = soup.find('div', {'class': 'commits-listing'})
li_l = div_ele.find_all('li', {'class': 'commits-list-item'})
for li in li_l:
    try:
        lst = []
        auth = parseText(li.find('a', {'class': 'commit-author'}).text)
        dt = parseText(li.find('relative-time').text)
        if tit_ele:
            cmt_title = parseText(tit_ele.text)
        lst.append(len(commit_data) + 1)
        lst.append(auth)
        commit_data.append(lst)    author, data)
    except:
        print("error")
        pass    print( commit_data )
```

The Listing 1 show how BeautifulSoup4 module parsing the HTML document, and crawling the website (commit data), and those code are the overall pattern in order to handle exceptions and make it coherent at the same time.

4.3.1. Result of the Research

The result of this research is a web-based application of monitoring task system integrated with GitHub are lecturer can get the information on repository, collaborators, commits and issues as shown at Figure 3 which can be shorten in the process of calculating the level of students' activeness according to their contribution that have been made within a certain period.

A	B	C	D	E	F
S.No	Author Name	Date	Commit Title	Commit Link	Commit Description
1	awangga	Jan 18, 2018	indra maju	https://github.com/BukuInformatika/SIG/commit/b23075aca3e81f82e3089096b63112e556be878a	
2	awangga	Jan 18, 2018	Merge pull request #92 from KELOMPOKD4TI3A/r	https://github.com/BukuInformatika/SIG/commit/a387de8f9e4b1072fa261fe261aabc70be28dc86	Kelompok 3 Mapserver
3	indrariksa	Jan 18, 2018	:(	https://github.com/BukuInformatika/SIG/commit/fb44a16c8b55c6d78f4d83bb6a57fa5439dc20f0	
4	indrariksa	Jan 18, 2018	ref gambar	https://github.com/BukuInformatika/SIG/commit/27f273bef9ab71d5df63347ac5545e3f01f09110	
5	awangga	Jan 18, 2018	maul maju	https://github.com/BukuInformatika/SIG/commit/4eaa2325ca7817d86dfa8ee472264839 98fec735	
6	awangga	Jan 18, 2018	maul	https://github.com/BukuInformatika/SIG/commit/0af88a2eaa2fd8101d2362d4f84442101b84eca0	
7	awangga	Jan 18, 2018	Merge branch 'master' of github.com:BukuInform	https://github.com/BukuInformatika/SIG/commit/4e72110505125e34a33f7edadaf2b60c62663d04	
8	awangga	Jan 18, 2018	oke	https://github.com/BukuInformatika/SIG/commit/fb5fc899f307edb1052390772bb35c66c9d0d8c6	
9	awangga	Jan 18, 2018	Merge pull request #91 from MMIAS/master...	https://github.com/BukuInformatika/SIG/commit/8c2ece211ace5c5b2071ba82b679fc81c423750f	Fix Kelompok 2
10	Maulyanda	Jan 18, 2018	Merge branch 'master' into master	https://github.com/BukuInformatika/SIG/commit/efcdb5b2c669385fc8769fdf1cff15b85fd622e8	
11	Maulyanda	Jan 18, 2018	Merge branch 'master' into master	https://github.com/BukuInformatika/SIG/commit/16eb974f81806f2c7b9c5499e8ecd198626309a2	
12	Maulyanda	Jan 18, 2018	Add files via upload	https://github.com/BukuInformatika/SIG/commit/946d309ef9abd1d8b56370aaaf0f5b47daa95d9c	
13	awangga	Jan 18, 2018	tugas 3c	https://github.com/BukuInformatika/SIG/commit/0c991e0286d6fd6a90fe93bfdb869341fa422ddf	
14	berlinmitraa	Jan 9, 2018	Update MapserverBali.tex	https://github.com/BukuInformatika/SIG/commit/a78703ac570a17346ce2a2ae669cc33b1c3b60b1	
15	kindiherdiansyah	Jan 9, 2018	Update MapserverBali.tex	https://github.com/BukuInformatika/SIG/commit/4228a5cf056fe3771f44fbf3e75e0bca3a8310e8	
16	indrariksa	Jan 9, 2018	:D	https://github.com/BukuInformatika/SIG/commit/898b115241202675 51b63ee79e0bdd449ea3c461	
17	ilgaanne	Jan 8, 2018	Update singapore.tex	https://github.com/BukuInformatika/SIG/commit/ede3fa1e4cf87f07dbaa9881ff26380bfcce995b	
18	indrariksa	Jan 8, 2018	au ah	https://github.com/BukuInformatika/SIG/commit/3ed9b253b1bf11bdef3381a04f043de2f3bf463f	
19	awangga	Jan 8, 2018	kelas B	https://github.com/BukuInformatika/SIG/commit/5626644b0b38adddc429e0c8f437b1b302024185	
20	mefrinkazela	Jan 7, 2018	Update singapore.tex	https://github.com/BukuInformatika/SIG/commit/a4a5b4e9d12dacdc49d303e3286d0a2fc1715ddb	
21	ilgaanne	Jan 6, 2018	mapzen	https://github.com/BukuInformatika/SIG/commit/8bd5236274c1d24ef55cfb71a8fa21822096952b	
22	mefrinkazela	Jan 6, 2018	Add files via upload	https://github.com/BukuInformatika/SIG/commit/8503cb250426a6491754064feac585742fcebe64	
23	mefrinkazela	Jan 6, 2018	apa aja	https://github.com/BukuInformatika/SIG/commit/183a8ca922c872ddcdb234890cd75e3520731de1	
24	Maulyanda	Jan 5, 2018	Add files via upload	https://github.com/BukuInformatika/SIG/commit/63047a9785622 5a2f9f3f2bcd2181947af5d1262	

Figure 3. The Result of Data Extraction

4.4. Maintenance

Maintenance phase is the last phase of waterfall model SDLC method [15]. In this phase, the author do the maintenance of the application. In case if there any changes in the design structure of GitHub, the data scraping process which is conducted before using the same pattern will be failed to obtain. This failure can be detected if there is an indication the data that obtained in scraping data process are decreased or the scraping pattern cannot obtain data at all. If any of this condition happen, the author need to reanalyze on every stage of data retrieval so it can be discovered which stage that has to be changed.

4.5. Response Time

This is the crucial one, how fast the system to be use in e.g a web service. Response time is the total amount of time it takes to respond to a request for service. The response time is the sum of the service time and wait time [16]. The service time is the time it takes to do the work author requested. The wait time is how long the request had to wait in a chain before being serviced. For this research, the response time for each URL shown by Table 2 below, meanwhile, if we run all the URL at the same time, it took 23.2 seconds to response.

Table 2. Response Time

URL	Response time
https://github.com/bukuinformatika/sig	8.1s
https://github.com/bukuinformatika/sig/issues	7.4s
https://github.com/bukuinformatika/sig/commits?author=awangga	7.7s

5. Conclusion

After performing the analysis, the implementation of web scraping in monitoring tasks integrated with GitHub, it can be concluded that the built application has been able to answer the problems discussed in the previous chapters. Our work shows that with the design of the system facilitate data collection tasks using social networking media GitHub, documentation and collection of tasks more structured.

In this research showed that the lecturer can get information on GitHub: repository details; collaborators details; commits count; detailed issues of the repository which can be shorten in the process of calculating the level of students' activeness according to their contribution that have been made within a certain period at one course.

References

[1] M. J. Hannafin, "Student-centered learning," in *Encyclopedia of the Sciences of Learning*. Springer, 2012, pp. 3211–3214.
[2] S. Armiati and R. Awangga, "Sql collaborative learning framework based on soa," in *Journal of Physics: Conference Series*, vol. 1007, no. 1. IOP Publishing, 2018, p. 012035.
[3] D. H. Jonassen and M. A. Easter, "Conceptual change and student-centered learning environments," *Theoretical foundations of learning environments*, pp. 95–113, 2012.
[4] N. Dabbagh and A. Kitsantas, "Personal learning environments, social media, and self-regulated learning: A natural formula for connecting formal and informal learning," *The Internet and higher education*, vol. 15, no. 1, pp. 3–8, 2012.
[5] E. Kalliamvakou, G. Gousios, K. Blincoe, L. Singer, D. M. German, and D. Damian, "The promises and perils of mining github," in *Proceedings of the 11th working conference on mining software repositories*. ACM, 2014, pp. 92–101.
[6] J. Feliciano, M.-A. Storey, and A. Zagalsky, "Student experiences using github in software engineering courses: a case study," in *Proceedings of the 38th International Conference on Software Engineering Companion*. ACM, 2016, pp. 422–431.

[7] A. Zagalsky, J. Feliciano, M.-A. Storey, Y. Zhao, and W. Wang, "The emergence of github as a collaborative platform for education," in *Proceedings of the 18th ACM Conference on Computer Supported Cooperative Work & Social Computing*. ACM, 2015, pp. 1906–1917.
[8] L. R. Julian and F. Natalia, "The use of web scraping in computer parts and assembly price comparison," in *New Media (CONMEDIA), 2015 3rd International Conference on*. IEEE, 2015, pp. 1–6.
[9] K. Sundaramoorthy, R. Durga, and S. Nagadarshini, "Newsone—an aggregation system for news using web scraping method," in *Technical Advancements in Computers and Communications (ICTACC), 2017 International Conference on*. IEEE, 2017, pp. 136–140.
[10] S. K. Malik and S. Rizvi, "Information extraction using web usage mining, web scrapping and semantic annotation," in *Computational Intelligence and Communication Networks (CICN), 2011 International Conference on*. IEEE, 2011, pp. 465–469.
[11] R. Mitchell, *Web scraping with Python: collecting data from the modern web*. " O'Reilly Media, Inc.", 2015.
[12] M. Grinberg, *Flask web development: developing web applications with python*. " O'Reilly Media, Inc.", 2014.
[13] G. Wilcock, "Pipelines, templates and transformations: Xml for natural language generation," in *Proceedings of the 1st NLP and XML Workshop*, 2001, pp. 1–8.
[14] V. G. Nair, *Getting Started with Beautiful Soup*. Packt Publishing Ltd, 2014.
[15] M. Mahalakshmi and M. Sundararajan, "Traditional sdlc vs scrum methodology–a comparative study," *International Journal of Emerging Technology and Advanced Engineering*, vol. 3, no. 6, pp. 192–196, 2013.
[16] R. Jain, *The art of computer systems performance analysis: techniques for experimental design, measurement, simulation, and modeling*. John Wiley & Sons, 1990.