ISSN : 2302-3805

ELICITING DATA FROM WEBSITE USING SCRAPY: AN EXAMPLE

Amna Shifia Nisafani[1)], **Rully Agus Hendrawan**[2)], **Arif Wibisono**[3)]

Sistem Informasi, Institut Teknologi Sepuluh Nopember
Jl Raya ITS, Sukolilo, Surabaya
Email : 1[1)]*, ruhendrawan@gmail.com*[2)]*, arif.wibisono@gmail.com*[3)]

Abstract

Data elicitation is one of the most crucial phases in any research. One of the methods that can be employed to gather data is web crawling. This paper aims to provide an example on how to elicit data from website using Scrapy in Windows Environment which has not been addressed in any previous resources. The result shows that Scrapy can be utilized to extract dan store data from any website.

Keywords: *data elicitation, scrapy, web crawling*

1. Introduction

Data elicitation is any kind of activities that aims to gather data from many sources [1]. Furthermore, data elicitation is one of the most crucial stages in any research [2]. Data elicitation is also known as data gathering [1], or data collection [3]. Data can be collected through several methods such as conducting experiment [3], social survey [4], interview [5] and focus group discussion [6]. All of the aforementioned methods are traditional and are utilized to extract data from primary sources. However, these traditional methods can be extended by the means of the internet as the abundant amount of user-generated data are now available.

A widely used method to automatically extract data from the internet is web crawling. Web crawling is a process of gathering web pages and indexing the pages [7,8]. By using web crawling, it will be quicker and more effective in collecting as many useful web pages as possible, including the links structure that interconnect them [9]. A program that download pages automatically is known as web crawler or web spider [10].

There are a lot of softwares that have been developed to crawl data from web such as PDFtoExcelOnLine, Zamzar, CometDocs, PDFTables, Tabula, import.io, kimonolabs, myTrama, Mozenda, QuBole, ScraperWiki, Apache Nutch, and Scrapy [11]. Among those softwares, we select Scrapy as our main tool in extracting data. There are several reasons to do so. First, Scrapy can best to deal with commonly-encountered broken HTML [12]. Second, Scrapy has a strong community support, this support will be valuable in case one encounters problem in executing scrapy [12]. Third, since it is written in python programming language, one can expect simplicity in its line of code. Hence, this feature will certainly help developer to develop and maintain a more sophisticated crawling program.

There is an abundant resources to assist those who learn how to use python in real world application. However, these resources did not incorporate some problems encountered during installation process especially in windows platform. Therefore, the problem statement of this study is that how to implement scrapy to a real world application in Windows. Here, we focus on retrieving data from portal Garuda, the largest academic repository in Indonesia. The purpose of doing so is to initially develop a robust academic analytics to capture the research trend in Indonesia. The objective of this paper is to demonstrate the basic technique to scrape data in Portal Garuda. Several steps by steps are given in order to ease reader to understand the flow of the framework.

In the end, we hope that this study can assist anyone that has interest in researching how the academic landscapes are evolved in Indonesia. The reminder of the paper is organized as follows. In the section two, we will discuss the underlaying study literature of this study. Furthermore, in this section, we show the step by step procedure of using scrapy framework. In the end, we conclude our work in section three.

2. Discussion

Before we explain on how to use Scrapy, here are some introduction to Scrapy and its architecture.

2.1. Scrapy

Scrapy is a powerful tool to extract data from many different internet sources [12]. Scrapy is an open source software developed by a london-based company MyDeco. The salient feature of scrapy is its capabilities to extract and combine data from various unstructured data sources. The open-source-software Scrapy offers a built-in procedure to extract data (named “selectors”). However, one may use another approach called Beautiful Soup (or Ixml). The scrapy is widely used by major information technology companies such as Lyst, Career Builder, SciencePo Media Lab, Data.go.uk World Government Data site.

ISSN : 2302-3805

2.2. Scrapy Architecture

Scrapy has several components within (see Figure 1). The spider engine manages the data flow among all components within scrapy. It is also responsible to trigger specific action whenever certain events occur. Scheduler is responsible to receive request from the engine and manage the queue of request to the scrapy engine. Downloader administer the web pages fetchment and give the fetch to the scrapy engine. Spiders are a custom class to parse responses and extract data. The item pipeline is responsible to process the data once they are gathered by the spiders. At this point, several activities are performed by the item pipeline such as cleansing, validation, and store to the database.

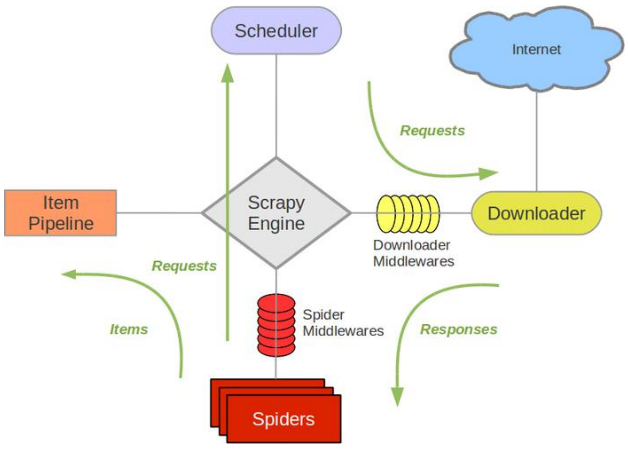


Figure 1. *Scrapy Architecture [13]*

2.3. Case Study

This section will explain about the use of Scrapy to elicit list of journals in Portal Garuda under Computer Science & IT Subject. The data source and data attributes have been defined in Table 1. The methodology employed in this research consists of four steps. First is defining data sources and data atrributes. The next step is to installing Scrapy in a computer/laptop. The third step is creating a spider project using Scrapy, and finally, the last step is to run the spider to get the data from the defined data sources.

2.3.1. Defining Data Source and Data Attributes

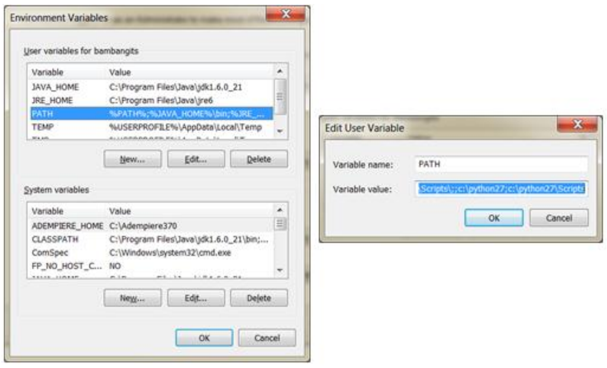
Suppose that we want to get some information from Portal Garuda on the list of journals in Computer Science & IT Subject, then the data source that will be used in this paper is portal garuda with the subject of Computer Science & IT (http://id.portalgaruda.org/?ref=browse&mod=area&area=60**}**) and the data attributes that will be collected are as the following table.

Table 1. List of attributes

ID	Attributes	Description
1.	Title	This attribute will describe the title of journal listed under Computer Science & IT subject in Portal Garuda
2.	Link	This attribute will describe the link of the journal listed under Computer Science & IT subject in Portal Garuda
3.	Publisher	This attribute will describe the publisher of the journal listed under Computer Science & IT subject in Portal Garuda

2.3.2 Installing Scrapy

In order to install Scrapy, there are several requirements that need to be address. These requirements will be explained as follow. First, since Scrapy is a python based framework, then the first requirement is to have python in our computer/laptop. Phyton can be downloaded through https://www.python.org/downloads/. It is important to note, that if windows laptop is used for data gathering, then it is suggested to download Scrapy 2.7. Another important point is that, we need to add new path where python has been installed. The path is ***C:\Python27\;C:\Python27\Scripts\;***. The illustration is depicted by Figure 2. Moreover, updating path should be done through windows command prompt as well by running the syntax ***"c:\python27\python.exe c:\python27\tools\scripts\win_add2path.py".*** Then**,** type ***python –version*** to check whether the path has already been updated. **Error! Reference source not found.**3 depicts the process of updating path in command prompt.



User variables for bambangits

Variable	Value
JAVA_HOME	C:\Program Files\Java\jdk1.6.0_21
JRE_HOME	C:\Program Files\Java\jre6
PATH	%PATH%;%JAVA_HOME%\bin;%JRE_...
TEMP	%USERPROFILE%\AppData\Local\Temp

System variables

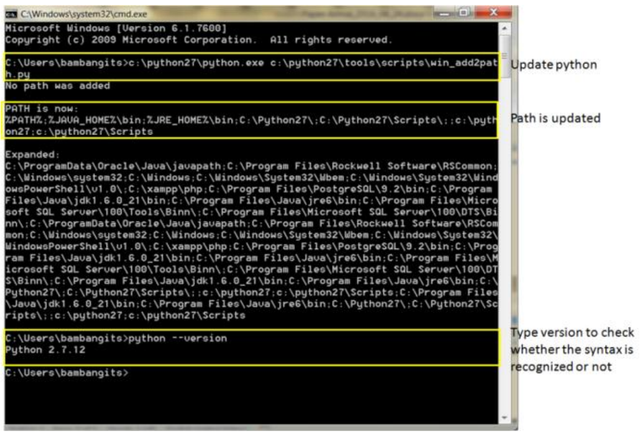
Variable	Value
ADEMPIERE_HOME	C:\Adempiere370
CLASSPATH	C:\Program Files\Java\jdk1.6.0_21\bin;...
ComSpec	C:\Windows\system32\cmd.exe
FP_NO_HOST_C...	NO

Variable name: PATH
Variable value: Scripts\;;c:\python27;c:\python27\Scripts

Figure 2. *Adding new path*

After python has been installed, the second requirement is to install pywin32 which is a set of extension modules that provides access to many of the Windows API functions [14]. The link to download pywin32 is https://sourceforge.net/projects/pywin32/files/pywin32/Build%20220/ . It is important to note that since we use python 2.7, the pywin32 that should be downloaded is ***pywin32-220.win32-py2.7.exe***.

ISSN : 2302-3805



```
Microsoft Windows [Version 6.1.7600]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Users\bambangits>c:\python27\python.exe c:\python27\tools\scripts\win_add2pat
h.py
No path was added

PATH is now:
%PATH%;%JAVA_HOME%\bin;%JRE_HOME%\bin;C:\Python27\;C:\Python27\Scripts\;;c:\pyth
on27;c:\python27\Scripts

Expanded:
C:\ProgramData\Oracle\Java\javapath;C:\Program Files\Rockwell Software\RSCommon;
C:\Windows\system32;C:\Windows;C:\Windows\System32\Wbem;C:\Windows\System32\Wind
owsPowerShell\v1.0\;C:\xampp\php;C:\Program Files\PostgreSQL\9.2\bin;C:\Program 
Files\Java\jdk1.6.0_21\bin;C:\Program Files\Java\jre6\bin;C:\Program Files\Micro
soft SQL Server\100\Tools\Binn\;C:\Program Files\Microsoft SQL Server\100\DTS\Bi
nn\;C:\ProgramData\Oracle\Java\javapath;C:\Program Files\Rockwell Software\RSCom
mon;C:\Windows\system32;C:\Windows;C:\Windows\System32\Wbem;C:\Windows\System32\
WindowsPowerShell\v1.0\;C:\xampp\php;C:\Program Files\PostgreSQL\9.2\bin;C:\Prog
ram Files\Java\jdk1.6.0_21\bin;C:\Program Files\Java\jre6\bin;C:\Program Files\M
icrosoft SQL Server\100\Tools\Binn\;C:\Program Files\Microsoft SQL Server\100\DT
S\Binn\;C:\Program Files\Java\jdk1.6.0_21\bin;C:\Program Files\Java\jre6\bin;C:\
Python27\;C:\Python27\Scripts\;;c:\python27;c:\python27\Scripts;C:\Program Files
\Java\jdk1.6.0_21\bin;C:\Program Files\Java\jre6\bin;C:\Python27\;C:\Python27\Sc
ripts\;;c:\python27;c:\python27\Scripts

C:\Users\bambangits>python --version
Python 2.7.12

C:\Users\bambangits>
```

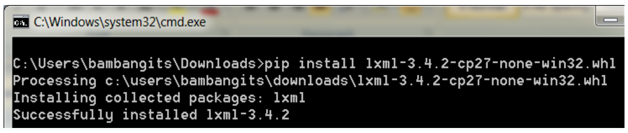
C:\Windows\system32\cmd.exe
Update python
Path is updated
Type version to check whether the syntax is recognized or not

Figure 3. *Updating path through windows command prompt*

The third requirement is to install Scrapy using ***pip install Scrapy*** syntax. Another point to note is that if we use proxy, then we have to set proxy using ***SET HTTPS_PROXY=username:password@proxy:port.*** Installing Scrapy in Windows need two configurations as the following table.

Table 2. *Configuration in Windows*

ID	Requirement	Configuration
1.	Python Compiler in Windows	Intall Microsoft Visual C++ Compiler for Python 2.7 through the following link http://www.microsoft.com/en-gb/download/details.aspx?id=44266
2.	Wheel	Download wheel through the following link https://pypi.python.org/simple/lxml/ and choose ***lxml-3.4.2-cp27-none-win32.whl*** . Then, install using comand prompt with the directory where the file stored as depicted in Figure 4. The syntax is ***pip install filename.whl***



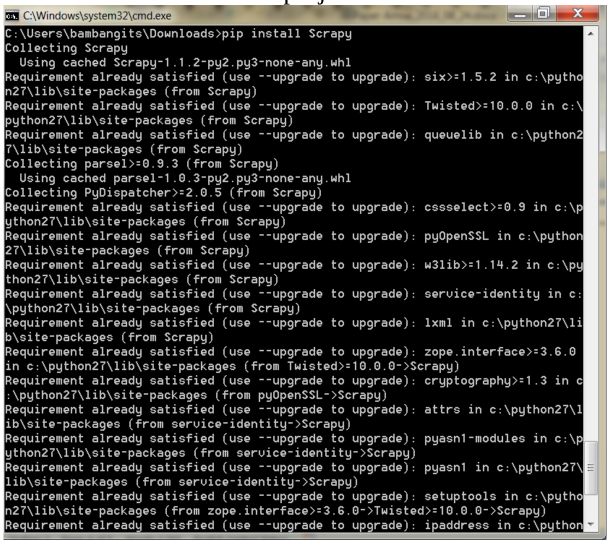
```
C:\Users\bambangits\Downloads>pip install lxml-3.4.2-cp27-none-win32.whl
Processing c:\users\bambangits\downloads\lxml-3.4.2-cp27-none-win32.whl
Installing collected packages: lxml
Successfully installed lxml-3.4.2
```

Figure 4. *Installing wheel*

Failing to configure these two requirement will cause problem in installing Scrapy. Figure 5 shows how to install Scrapy in comand prompt.

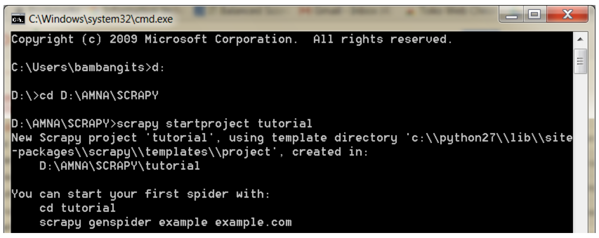
2.3.4. Creating A Spider Project

After Scrapy has been installed, the next step is to create a spider project. A spider which is known as web crawler is a program that will download data from web pages. In order to create project using Scrapy, first, we need to define the directory in which the code will be stored. Then in the directory, we can run ***scrapy startproject project'sname.*** The illustration can be seen in **Error! Reference source not found.**. In this case, we use tutorial as the name of the project.



```
C:\Users\bambangits\Downloads>pip install Scrapy
Collecting Scrapy
  Using cached Scrapy-1.1.2-py2.py3-none-any.whl
Requirement already satisfied (use --upgrade to upgrade): six>=1.5.2 in c:\pytho
n27\lib\site-packages (from Scrapy)
Requirement already satisfied (use --upgrade to upgrade): Twisted>=10.0.0 in c:\
python27\lib\site-packages (from Scrapy)
Requirement already satisfied (use --upgrade to upgrade): queuelib in c:\python2
7\lib\site-packages (from Scrapy)
Collecting parsel>=0.9.3 (from Scrapy)
  Using cached parsel-1.0.3-py2.py3-none-any.whl
Collecting PyDispatcher>=2.0.5 (from Scrapy)
Requirement already satisfied (use --upgrade to upgrade): cssselect>=0.9 in c:\p
ython27\lib\site-packages (from Scrapy)
Requirement already satisfied (use --upgrade to upgrade): pyOpenSSL in c:\python
27\lib\site-packages (from Scrapy)
Requirement already satisfied (use --upgrade to upgrade): w3lib>=1.14.2 in c:\py
thon27\lib\site-packages (from Scrapy)
Requirement already satisfied (use --upgrade to upgrade): service-identity in c:
\python27\lib\site-packages (from Scrapy)
Requirement already satisfied (use --upgrade to upgrade): lxml in c:\python27\li
b\site-packages (from Scrapy)
Requirement already satisfied (use --upgrade to upgrade): zope.interface>=3.6.0 
in c:\python27\lib\site-packages (from Twisted>=10.0.0->Scrapy)
Requirement already satisfied (use --upgrade to upgrade): cryptography>=1.3 in c
:\python27\lib\site-packages (from pyOpenSSL->Scrapy)
Requirement already satisfied (use --upgrade to upgrade): attrs in c:\python27\l
ib\site-packages (from service-identity->Scrapy)
Requirement already satisfied (use --upgrade to upgrade): pyasn1-modules in c:\p
ython27\lib\site-packages (from service-identity->Scrapy)
Requirement already satisfied (use --upgrade to upgrade): pyasn1 in c:\python27\
lib\site-packages (from service-identity->Scrapy)
Requirement already satisfied (use --upgrade to upgrade): setuptools in c:\pytho
n27\lib\site-packages (from zope.interface>=3.6.0->Twisted>=10.0.0->Scrapy)
Requirement already satisfied (use --upgrade to upgrade): ipaddress in c:\python
```

Figure 5. *Installing scrapy*



```
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Users\bambangits>d:

D:\>cd D:\AMNA\SCRAPY

D:\AMNA\SCRAPY>scrapy startproject tutorial
New Scrapy project 'tutorial', using template directory 'c:\\python27\\lib\\site
-packages\\scrapy\\templates\\project', created in:
    D:\AMNA\SCRAPY\tutorial

You can start your first spider with:
    cd tutorial
    scrapy genspider example example.com
```

Figure 6. *Creating scrapy project*

When the project is created, new folder is created in the predefined directory. The folder's name is equal to the project's name. Inside the folder, some python codes have been generated as depicted by **Error! Reference source not found.**. Description for each content can be seen in Table 3.

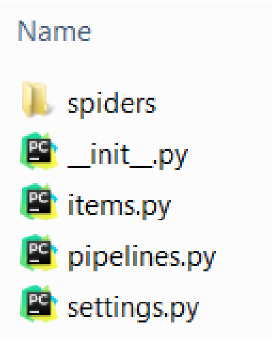


Figure 7. *Contents of the project*

Table 3. *Project Content Description*

ID	Content	Description
1.	items.py	project items file
2.	pipelines.py	project pipelines file
3.	settings.py	project settings file
4.	spiders/	a directory where you'll later put your spiders

The first thing that should be edited is items.py file. This file is used to define the items of data that will be

ISSN : 2302-3805

collected. In our example, the items will be equal to data attributes defined in Table 1. The code for items.py can be seen in **Error! Reference source not found.**.

```
1     class     JournalsItem(scrapy.Item):
2      # define the fields for your item
here                                like:
3             #   name   =   scrapy.Field()
4               title    =   scrapy.Field()
5    link =  scrapy.Field()
6          publisher   =   scrapy.Field()
7    pass
```

Figure 8. *code for item.py*

In Figure 8 on the first row, we define the name of the class. Here, we use JournalsItem. This class will be called to store data as an object. More details, we state our attributes, namely "*title*", "*link*", and "*publisher*" as an item in row 4 and 5.

The second thing that should be configured is a scrapy spider. This spider can be created using python editor. The spider class is situated in the folder spiders. In this class, we defined list of URLs that will be mined to gather some data and information. In order to get the data, we need to define selector. The selectors for each data item are stated in the following table.

Table 4. *List of selectors*

ID	Data Item	Selector
1.	Title	//td//div [@class="item_title"]//a//text()
2.	Link	//td//div [@class="item_title"]//a//@href
3.	Publisher	//td//div [@class="item_publisher"].text()

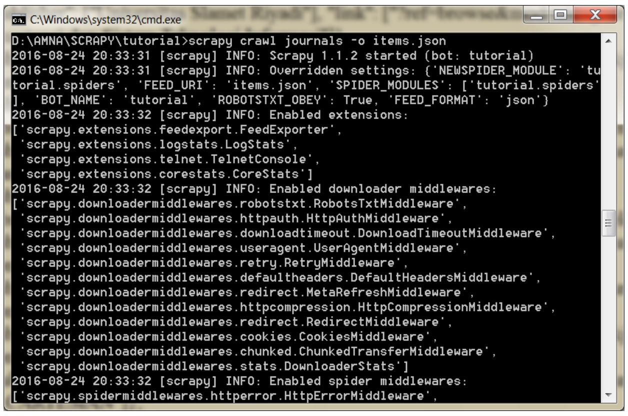
Using these selectors, we then configure spider file as depicted in Figure 9. We name the class as journals_spider.py. As seen from Figure 9, we import JournalsItem that have been defined in Figure 8 to store the collected data (row 2). Furthormore, we define the name of spider, domain of web page, and URLs in row 5, 6 and 7 respectively. Ultimately, we get data by setting the xpath selector defined in Table 4 in row 14 to 19.

2.3.5. Run Spider

When the spider has been developed, the subsequent step is to run the script using comand prompt. Th syntax is ***scrapy crawl spider'sname –o items.csv***. ***Scrapy crawl*** is the syntax to run spider. Syntax ***–o items.json*** will store the data into json file, while spider's name is the name of spides. In this case, the name of the spider is journals. **Error! Reference source not found.** shows how to run the spider and **Error! Reference source not found.** depicts the captured data in json file.

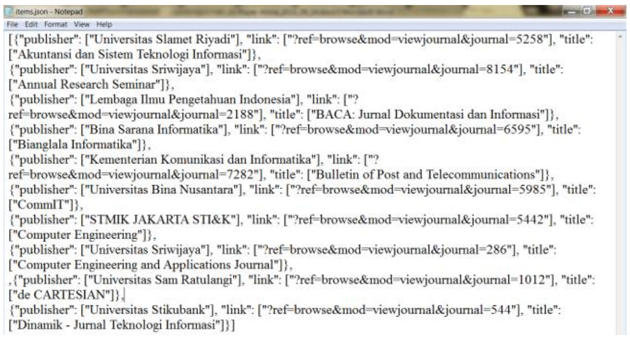
```
1  import scrapy
2  from tutorial.items import
JournalsItem
3
4  class JournalsSpider(scrapy.Spider):
5      name = "journals"
6      allowed_domains =
["http://id.portalgaruda.org/"]
7      start_urls = [
8
"http://id.portalgaruda.org/?ref=browse&m
od=area&area=60"
9      ]
10
11     def parse(self, response):
12         for sel in
response.xpath('//td'):
13             item = JournalsItem()
14             item['title'] =
15
sel.xpath('div[@class="item_title"]//a/te
xt()').extract()
16             item['link'] =
17
sel.xpath('div[@class="item_title"]//a//@
href').extract()
18             item['publisher'] =
19
sel.xpath('div[@class="item_publisher"]/t
ext()').extract()
20                 yield item
```

Figure 9. *Code for spider*



```
D:\AMNA\SCRAPY\tutorial>scrapy crawl journals -o items.json
2016-08-24 20:33:31 [scrapy] INFO: Scrapy 1.1.2 started (bot: tutorial)
2016-08-24 20:33:31 [scrapy] INFO: Overridden settings: {'NEWSPIDER_MODULE': 'tu
torial.spiders', 'FEED_URI': 'items.json', 'SPIDER_MODULES': ['tutorial.spiders'
], 'BOT_NAME': 'tutorial', 'ROBOTSTXT_OBEY': True, 'FEED_FORMAT': 'json'}
2016-08-24 20:33:32 [scrapy] INFO: Enabled extensions:
['scrapy.extensions.feedexport.FeedExporter',
 'scrapy.extensions.logstats.LogStats',
 'scrapy.extensions.telnet.TelnetConsole',
 'scrapy.extensions.corestats.CoreStats']
2016-08-24 20:33:32 [scrapy] INFO: Enabled downloader middlewares:
['scrapy.downloadermiddlewares.robotstxt.RobotsTxtMiddleware',
 'scrapy.downloadermiddlewares.httpauth.HttpAuthMiddleware',
 'scrapy.downloadermiddlewares.downloadtimeout.DownloadTimeoutMiddleware',
 'scrapy.downloadermiddlewares.useragent.UserAgentMiddleware',
 'scrapy.downloadermiddlewares.retry.RetryMiddleware',
 'scrapy.downloadermiddlewares.defaultheaders.DefaultHeadersMiddleware',
 'scrapy.downloadermiddlewares.redirect.MetaRefreshMiddleware',
 'scrapy.downloadermiddlewares.httpcompression.HttpCompressionMiddleware',
 'scrapy.downloadermiddlewares.redirect.RedirectMiddleware',
 'scrapy.downloadermiddlewares.cookies.CookiesMiddleware',
 'scrapy.downloadermiddlewares.chunked.ChunkedTransferMiddleware',
 'scrapy.downloadermiddlewares.stats.DownloaderStats']
2016-08-24 20:33:32 [scrapy] INFO: Enabled spider middlewares:
['scrapy.spidermiddlewares.httperror.HttpErrorMiddleware',
```

Figure 10. *Running Spider*



```
[{"publisher": ["Universitas Slamet Riyadi"], "link": ["?ref=browse&mod=viewjournal&journal=5258"], "title":
["Akuntansi dan Sistem Teknologi Informasi"]},
{"publisher": ["Universitas Sriwijaya"], "link": ["?ref=browse&mod=viewjournal&journal=8154"], "title":
["Annual Research Seminar"]},
{"publisher": ["Lembaga Ilmu Pengetahuan Indonesia"], "link": ["?
ref=browse&mod=viewjournal&journal=2188"], "title": ["BACA: Jurnal Dokumentasi dan Informasi"]},
{"publisher": ["Bina Sarana Informatika"], "link": ["?ref=browse&mod=viewjournal&journal=6595"], "title":
["Bianglala Informatika"]},
{"publisher": ["Kementerian Komunikasi dan Informatika"], "link": ["?
ref=browse&mod=viewjournal&journal=7282"], "title": ["Bulletin of Post and Telecommunications"]},
{"publisher": ["Universitas Bina Nusantara"], "link": ["?ref=browse&mod=viewjournal&journal=5985"], "title":
["CommIT"]},
{"publisher": ["STMIK JAKARTA STI&K"], "link": ["?ref=browse&mod=viewjournal&journal=5442"], "title":
["Computer Engineering"]},
{"publisher": ["Universitas Sriwijaya"], "link": ["?ref=browse&mod=viewjournal&journal=286"], "title":
["Computer Engineering and Applications Journal"]},
,{"publisher": ["Universitas Sam Ratulangi"], "link": ["?ref=browse&mod=viewjournal&journal=1012"], "title":
["de CARTESIAN"]},
{"publisher": ["Universitas Stikubank"], "link": ["?ref=browse&mod=viewjournal&journal=544"], "title":
["Dinamik - Jurnal Teknologi Informasi"]}]
```

Figure 11. *Result of web crawling*

3. Conclusion

This paper provides an example on how to elicit data from website using Scrapy. This paper also highlights some problems that should be address when setting

Scrapy in Windows environment which did not incorporated in any tutorial resources and documentation. We use a case study of journal listing from Portal Garuda. From the example, it can be shown that Scrapy can be easly utilized in terms of simplicity of the code. Further research will focus on more advanced setting especially dealing with nested webpages and heavily contained with JavaScript.

References

[1] Deakin University. (2007) [Online]. http://www.deakin.edu.au/__data/assets/pdf_file/0006/271527/data-gathering1.pdf
[2] Peersman, G. (2014). *Methodological Brief No.10: Overview: Data Collection and Analysis Methods in Impact Evaluation.* Florence, Italy: UNICEF Office of Research - Innocenti
[3] Joop J. Hox and Hennie R. Boeji, "Data Collection, Primary vs Secondary," Encyclopedia of Social Measurement, vol. 1, pp. 593-599, 2005.
[4] Edith D. de Leeuw, Joop J. Hox, and Don A. Dillman, International Handbook of Survey Methodology.: European Association of Methodology, 2008.
[5] Jody Miller and Barry Glassner, "The Inside and the Outside : Finding Realities in Interviews," in Qualitative Research, David Silverman, Ed. London: Sage, 2016, pp. 51-66.
[6] Sue Wilkinson, "Analysing Focus Group Data," in Qualitative Research, David Silverman, Ed. London: Sage, 2016, pp. 83-100.
[7] Vassiliki Hatzi, B. Barla Cambazoglu, and Iordanis Koutsopoulos, "Optimal Web Page Download Scheduling Policies for Green Web Crawling," IEEE Journal on Selected Areas in Communication, vol. 34, no. 5, pp. 1378 - 1388, 2016.
[8] Christopher Olston and Marc Najork, "Web Crawling," Foundation and Trends in Information Retreival, vol. 4, no. 3, pp. 175-246, 2010.
[9] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schutze, Introduction to Information Retrieval.: Cambridge University Press, 2008.
[10] Pavel Pecina et al., "Domain adaptation of statistical machine translation with domain-focused web crawling," Langauge Resources and Evaluation, vol. 49, no. 1, pp. 147-193, 2015.
[11] Osmar Castrillo-Fernández, "Web Scraping: Applications and Tools," 2015.
[12] Dimitrios Kouzis-Loukas, Learning Scrapy, Learn the art of efficient web scraping and crawling. Birmingham: Packt Publishing Ltd., 2016.
[13] Scrapy Documentation [Online] http://doc.scrapy.org/en/latest/topics/architecture.html. *Accessed : November 2016*
[14]Dale Athanasias. (2014) Python Wiki. [Online].

About the Authors

Amna Shifia Nisafani, obtained her bachelor degree from Information Systems Department of Institut Teknologi Sepuluh Nopember Surabaya, graduated in 2007. She also finished her master degree from Institut of Logisctics Information Technology, Pusan National University, Korea, in 2011. Currently, she is working as a lecturer in Information Systems Departement of ITS Surabaya.

Rully Agus Hendrawan, obtained his bachelor degree from Informatics Department of Institut Teknologi Sepuluh Nopember Surabaya, and master degree from Retsumeikan University, Jepang. Currently, he is working as a lecturer in Information Systems Departement of ITS Surabaya.

Arif Wibisono, obtained his bachelor degree from Information Systems Department of Institut Teknologi Sepuluh Nopember Surabaya, graduated in 2007. He also finished his master degree from Institut of Logisctics Information Technology, Pusan National University, Korea, in 2011. Currently, he is working as a lecturer in Information Systems Departement of ITS Surabaya.

ISSN : 2302-3805