Automated Black Box Testing using Selenium Python

Nurul Rita Mustika[1] and Novrina[2]

[1, 2] Department of Computer Science and Information Technology, Universitas Gunadarma, Depok, Indonesia

[1]nrita107@gmail.com, [2]novrina@staff.gunadarma.ac.id

ABSTRACT

Software testing is important in software development flow because there is no guarantee that every software created has no bugs and errors. There are several types of testing software, one of them is black box testing that performs functional testing and in its development, this can be done automatically. Many automated testing tools are widely known, one of them is Selenium Python. To create an automated testing using Selenium Python, there are several steps such as planning, analysis, test design, converting test cases into automated tests, perform tests, and analyzed the test results.

Keywords: *Automated testing, Black Box Testing, Selenium Python, Testing*.

1. INTRODUCTION

Technological developments affect all aspects, including website growth. According to the Internet Live Stats website, there are more than 1 billion websites in the world today. This acquisition was first achieved in September 2014, which was confirmed by NetCraft, a Web Server Survey in October 2014. The amount then dropped to levels below 1 billion (due to monthly fluctuations in the count of inactive websites) before reaching back and stabilizing above 1 billion starting in March 2016 [1]. This affects website developers to keep up to date so that their website remains active and is visited by internet users. Therefore, testing is needed to find and correct errors when adding or updating certain modules.

In System Development Life Cycle (SDLC), software testing is executing a program or system to look for errors on a program or system. According to Roger [2] the goal of testing is to find and fix as many errors as possible in the program, before submitting the program to the customer. One of a good test is a test that has a high probability of finding errors.

Software testing is divided into 3, they are black box testing, white box test, and grey box testing [3]. Black box testing is a testing based on functional (UI experience), white box testing is a testing based on the code of system, and grey box testing is a testing that adopts black box and white box testing. Besides, software testing has 2 types in its process, they are manual testing and automated testing [4]. Manual testing is a testing that does not use automated tools and scripts, so testers take over the role of end users and test the software to identify unexpected behavior or bugs. Automated Testing is a testing that uses tools and scripts to perform testing and it can re-run test scenarios repeatedly, quickly, accurately, and save time and money when it compared with manual testing. There are several tools for automated testing, such as Selenium, Appium, Cucumber, Katalon Studio, Test Complete, etc.

2. LITERATURE REVIEW

2.1 Software Development Life Cycle (SDLC)

Software Development Life Cycle (SDLC) is concept used in software engineering to describe a procedure for planning, creating, coding, testing and implementation of user requirement specification. The systems development life-cycle applies to a range of hardware and software configurations [5].

2.2 Software Testing

In the system development flow, testing is done after the coding process, but testing is not only seen as the last step of the system development flow, testing has its own flow [6]. According to Myers [4], testing is a destructive process which explains why most people find it difficult. That may go against their grain because most people are inclined toward making objects rather than ripping them apart. Through the testing, it can be seen that there is no system or program that has no bug or error and testing also cannot guarantee no more errors. It can be affected by the addition of new modules, mod modules, and module removal.

2.2.1 Black Box Testing

In black box testing, or functional testing, test conditions are developed on basis of the program or system's functionality; that is, the tester requires information about the input data and observed output, but does not know how the program or system works. The tester focuses on testing program's functionality against the specification, hence the tester is completely unconcerned with the internal structure of the program or system [7].

2.2.1.1 Automated Testing

Automated testing requires writing up some special computer programs to find bugs or defects in software under test. It is an excellent approach to replace the laborious and time consuming manual testing [8]. Automated testing has various advantages and it is always
suggested for the quality improvements of the application as it provides formal test coverage, avoid human errors and speed up the test execution process [9]. Also, as it speeds up the execution process, it is the most effective solution for meeting the strict deadlines. There are several tools for automated testing, i.e. Selenium, Appium, Test Complete, Katalon Studio, Watir, Cucumber, Ranorex, etc.

2.3 Selenium

Selenium [10] is a robust set of tools that supports the rapid development of test automation for web-based applications. Selenium provides domain-specific testing languages (Selenese), for writing tests in a number of popular programming languages, including C #, Groovy, Java, Perl, PHP, Python, Ruby, and Scala. Testing can run on any web browser and can be done on Windows, Linux, and OS X platforms.

Selenium Python bindings provide a simple API to write functional/acceptance tests using Selenium WebDriver. Through Selenium Python API you can access all functionalities of Selenium WebDriver in an intuitive way.

Selenium Python bindings provide a convenient API to access Selenium Web-Drivers like Firefox, Internet Explorer, Chrome, Remote etc. The current supported Python versions are 2.7, 3.5 and above [11].

3. METHOD OF AUTOMATED TESTING

3.1 Research Method

In making automated black box testing using Selenium Python have several stages. The stages are illustrated in the following figure 1 below.

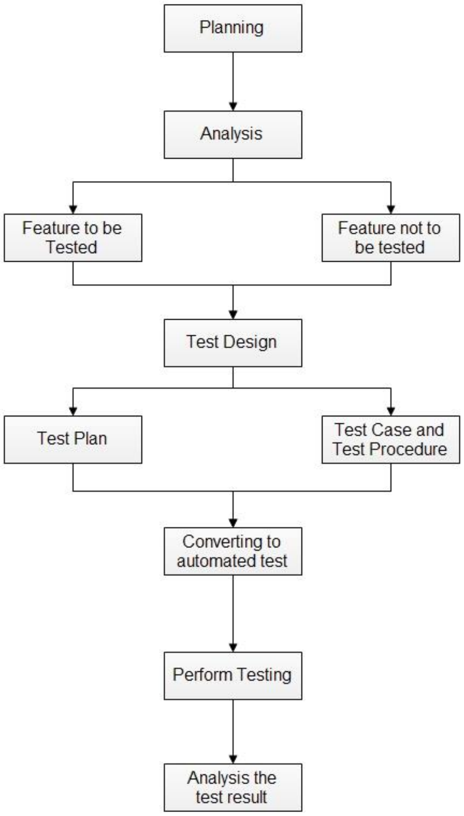


Fig. 1. Research Method

3.1.1 Planning

The initial stage is to make a plan by determining the object of research and research time. In this case, the research object is a website. Besides, determining the requirements are needed in doing the automated test using Selenium Python from the big things to the small things so that the scope of the test effort can be identified.

3.1.2 Analysis

In the analysis stage, is created a list of all features that exist in the research object. Then, made the selection of the features that will be tested and the features that will not be tested based on predetermined limits. In determining the features to be tested, the researcher tested the research object by using the black box testing manually to be used as a reference when creating automated black box testing using Selenium Python. Then an outline of automatic testing is performed to determine the features to be tested and features that will not be tested.

3.1.3 Test Design

The test design is divided into 2 sections, they are create the test plan and create test case and test procedure. Test plan will describe about the test hierarchy— in this case is navigation structure of research object, the item pass/fail criteria of the test, and suspension criteria and resumption requirement—laid-off testing activities due to something and to restart a certain way.

The test case will be filled by input (test item) and output of the test, and environmental requirement (hardware, software, and special requirement when doing the test). The test Procedure will be filled by the steps are taken to execute the test case.

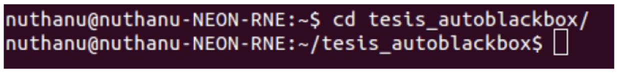
Input, output, and special requirement required are test cases as input, the expected result as output, and Selenium Python and internet connection as a special requirement. In using Selenium Python, it should install Selenium Python, put geckodriver to the testing folder, and add folder path to .bashrc by writing the command gedit ~/.bashrc in the terminal.

3.1.4 Converting to Automation Test Script

The all test cases will be converted into python script and it will be executed by using Selenium Python. Beforehand, Selenium Python must be installed and should be determined the browser that wanted to use. In this case, the browser for the test is Firefox.

3.1.5 Perform Testing

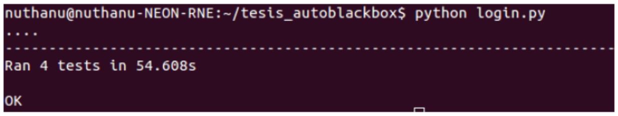
After converting all the test cases into python script, it will execute using Selenium Python through the terminal and Firefox as the browser. To execute the test, open the terminal and go to the folder test, as in figure 2.



```
nuthanu@nuthanu-NEON-RNE:~$ cd tesis_autoblackbox/
nuthanu@nuthanu-NEON-RNE:~/tesis_autoblackbox$
```

Fig. 2. Writing command in the terminal

Then select one of the test files, for example login.py—test case for login by write python login.py. The testing process will take place at some time and wait for it to complete, as in figure 3.



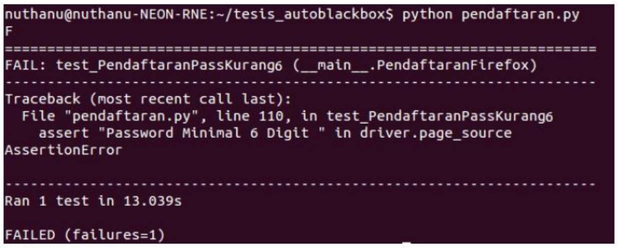
```
nuthanu@nuthanu-NEON-RNE:~/tesis_autoblackbox$ python login.py
....
----------------------------------------------------------------------
Ran 4 tests in 54.608s

OK
```

Fig. 3. Execute login.py

The figure above, explain that for running 4 tests (scenarios) in login.py need 54,608 seconds and “OK” means that the test is successful. The length of test time may vary depending on the internet connection, the time of the test, and the number of users that using the research object.

The test item is considered failed, if in the second line is written “F” and there is written “FAILED (failures=1)” that means a test case is failed not in accordance with the expected results). As in figure 4, In the fourth line, Fail:test_pendaftaranPassKurang6(__main__.PendaftaranFirefox) means the failed test occurred in function test_PendaftaranPassKurang6. The next line described the failed part (line code) was assert “Password Minimal 6 Digit” in driver.page_source and AssertionError means the driver did not find “Password Minimal 6 Digit” in this web, so error happened.



```
nuthanu@nuthanu-NEON-RNE:~/tesis_autoblackbox$ python pendaftaran.py
F
======================================================================
FAIL: test_PendaftaranPassKurang6 (__main__.PendaftaranFirefox)
----------------------------------------------------------------------
Traceback (most recent call last):
  File "pendaftaran.py", line 110, in test_PendaftaranPassKurang6
    assert "Password Minimal 6 Digit " in driver.page_source
AssertionError

----------------------------------------------------------------------
Ran 1 test in 13.039s

FAILED (failures=1)
```

Fig. 4. Result test item is considered failed

3.1.2 Analysis The Test Result

In this stage, after performing the test, the result of the tests are obtained. It will be gotten the successful/failed results of each test case that created before and will be analyzed according to the conclusions obtained from each test case. The analysis can be an evaluation for further development.

4. CONCLUSIONS

Software testing is important because the purpose of software testing is to find errors/bugs as many as possible in the system before the system is thrown to the consumer. In software testing, there are 3 types of testing, i.e. black box testing, white box testing, and grey box testing. In addition, the testing is divided into automated testing and manual testing. There are many devices for automated testing, one of them is Selenium Python. To create automated testing, could be through several stages, i.e. planning, analysis, test design, converting test cases into automated tests, perform test, and analyzed the test results. From the test result, it obtained are "success/failure" of each test case that has been made before and the length of test time when performing automatic testing. These results can be used as an evaluation for further development.

REFERENCES

[1] Internet Live Stats (2017). Internet Live Stats. [Online]. Available at http://www.internetlivestats.com/.

[2] Roger, S. P. (2012). Rekayasa perangkat lunak (pendekatan praktisi) edisi 7: Buku 1. Yogyakarta: Andi.

[3] Lewis, W. E. (2000). Software Testing and Continuous Quality Improvement.CRC Press.

[4] [4] Myers, G. J., Sandler, C., and Badgett, T. (2012). The Art of Software Testing. John Wiley & Sons, Inc, third edition edition.

[5] Shubhmeet Kaur, “A Review of Software Development Life Cycle Models”, International Journal of Advanced Research in Computer Science and Software Engineering, Volume 5, Issue 11, November 2015 ISSN: 2277 128X.

[6] Drabick, R. D. (2013). Best practices for the formal software testing process: A menu of testing tasks. Pearson Education.

[7] Lewis, W. E. (2000). Software Testing and Continuous Quality Improvement.CRC Press.

[8] Kumar, D. and Mishra, K. (2016). The impacts of test automation on software’s cost, quality and time to market. Procedia Computer Science, 79:8–15.

[9] Dustin, E. (2002). Effective Software Testing: 50 Ways to Improve Your Software Testing. Addison-Wesley Longman Publishing Co., Inc.

[10] Team, S. D. (2010). Selenium Documentation Release 1.0. [Online]. Available at: http://bit.ly/2oSj061.

[11] Muthukadan, B. (2017). Selenium Python Binding Release 2. [Online]. Available at: https://media.readthedocs.org/pdf/seleniumpython/latest/selenium-python.pdf.