**SURVEY PAPER: SOFTWARE AUTOMATED TESTING TOOL USING SYSTEMATIC LITERATURE REVIEW METHOD**

**Dheanda Absharina1; Fahirah2; Fenni Agustina3**

Master of Business Information Systems

Technology and Engineering Master Program

Gunadarma University

www.gunadarma.ac.id

[dheandaa@gmail.com](mailto:dheandaa@gmail.com)1, [eva190198@gmail.com](mailto:Eva190198@gmail.com)2, [fenni@staff.gunadarma.ac.id](mailto:fenni@staff.gunadarma.ac.id)3

**Abstract**— Programming testing is one of the most significant parts in fruitful programming advancement. The expanding unpredictability of programming improvement requires the advancement group to utilize mechanized testing apparatuses to test the quality and usefulness of the application. In programming testing, picking a testing instrument must be proper and as per the product to be tried. Using the Systematic Literature Review method, this research collects and analyzes previous survey papers from 2 keywords, "Comparative of Automated Software Testing Tools" and "A Critical Analysis of Automated Software Testing Tools" on Google Scholar with a span of 2010 - 2020. Results from papers collected through 5 stages of the SLR, resulting in 11 papers that were reviewed. From these 11 journals, obtained 5 automated testing tools that are often discussed, Selenium, QTP, TestCompelete, Watir, and Ranorex. The results of this study indicate that there is no software automatic testing tool that is truly perfect because every automated software testing tool has its own strengths and weaknesses and it is expected that the results of this study can help software testers in choosing automatic testing tools according to their needs.

**Keywords:** Automated Testing Tools, Software Testing, Systematic Literature Review

**Abstrak***—* Pengujian pemrograman adalah salah satu bagian terpenting dalam kemajuan pemrograman yang berhasil. Peningkatan ketidakpastian pemrograman yang semakin meluas membutuhkan kelompok pengembang untuk menggunakan peralatan pengujian mekanis untuk menguji kualitas dan kegunaan aplikasi. Dalam pengujian pemrograman, memilih instrumen pengujian harus tepat dan sesuai dengan produk yang akan dicoba.. Dengan menggunakan metode *Systematic Literatur Review* penelitian ini mengumpulkan dan menganalisis *survey paper* terdahulu dari 2 kata kunci, yaitu " *Comparative of Automated Software Testing Tools* " dan " *A Critical Analysis of Automated Software Testing Tools* " pada Google Scholar dengan rentang waktu 2010 – 2020. Hasil dari *paper* yang dikumpulkan melalui 5 tahapan SLR, menghasilkan 11 *paper* yang di *review*. Dari 11 jurnal tersebut diperoleh 5 alat uji yang sering dibahas, yaitu Selenium, QTP, Test Compelete, Watir, dan Ranorex. Hasil penelitian ini menunjukkan bahwa tidak ada alat pengujian otomatis perangkat lunak yang benar-benar sempurna, karena setiap alat pengujian otomatis perangkat lunak memiliki kelebihan dan kelemahannya masing-masing dan diharapkan hasil penelitian ini dapat membantu penguji perangkat lunak dalam memilih alat pengujian otomatis sesuai dengan kebutuhannya.

***Kata Kunci****:* *Alat Pengujian Otomatis, Pengujian Perangkat Lunak, Systematic Literature Review*

**INTRODUCTION**

Software testing is one of the most important processes in software development. According to (Malik & Gahlan, 2016), testing is a major component of any software engineering process which is meant to produce high quality applications. This test aims to ensure that the software developed meets all functional requirements, user needs, and is free from all forms of defects and errors (Okezie, Odun-Ayo, & Bogle, 2019). In other words, this test plays an important role in the success of software development.

Software testing can be done by manual testing or automated testing (Shaukat et al., 2015). Manual testing is performed by testers to manually test software for defects or errors (Malve & Sharma, 2017). In manual testing, the tester acts as the end user to test the behavior of almost all software features, so manual testing can be time-consuming and demanding, and does not always eliminate all bugs effectively. (Gamido & Gamido, 2019). Apart from being a very time-consuming process, manual testing is also unusable, lacks scripting facilities, requires considerable effort, and some errors remain uncovered. (Malik & Gahlan, 2016), whereas automated is carried out using software testing tools that can make it easy for software testers to automate the software testing process (Malve & Sharma, 2017), so that it is considered more effective in terms of time, cost, and usability. In addition, the test scenarios created by software testers in automated testing can be reused or re-executed. This reason has led most testers to switch from manual testing to automated software testing.

The expanding multifaceted nature of programming improvement requests that product advancement groups utilize robotized programming testing instruments to test application quality and usefulness. There are an assortment of devices accessible for mechanized testing, both open source and business, and there are programming trying instruments that just play out particular sorts of testing and are restricted to specific kinds of dialects (Gamido & Gamido, 2019). One of the most widely used open source automated testing tools, namely Selenium, is a combination of various automated testing tools (Mahmood & Sirshar, 2017). According to (Shalini & Thakur, 2017), the tool used for testing depends on the response time factor, the type of test, throughput, memory and CPU usage, language support, browser support, etc. This indicates that software testers must determine or select software testing tools appropriately and in accordance with the software to be tested.

According to the results of research conducted by (H. Kaur & Gupta, 2013), Quick Test Professional (QTP) is a software automated testing tool when compared to Selenium and TestComplete. However, if the test automation requirements are met with TestComplete, there is no need to use QTP which requires higher costs. Meanwhile, if you don't want to spend money on testing tools, Selenium can be used as an automated software testing tool.

Research conducted by (M. Kaur & Kumari, 2011), QTP and TestComplete are excellent tools for automated testing of software. In determining which to use between QTP and TestComplete, it can be determined based on the features of the application and the scope of the test. TestComplete has an easy-to-use UI and efficient playback, and is better used for applications with lower security requirements, whereas QTP is the best tool if data security is required even when testing.

(Singh & Tarika, 2014) evaluated the three most popular open source software, namely Selenium, Sikuli, and Watir. His research results show that Selenium ranks first, due to its enhanced recording features, data-driven testing, and ease of learning, improved support for third-party application integration. Watir ranks second, due to the lack of support for recording test cases removed by third-party applications, and lacks support for the languages that test cases can export while execution time is less than Selenium, while Sikuli ranks third, because it has the least recording capability, but has faster execution speed than selenium and Watir, and supports a variety of languages that can test test cases.

The SLR method is a method that can be used to conduct research on the review of a topic from various journals. The SLR method is useful for identifying, analyzing, assessing and describing all research related to an interesting theme and the appropriate questions from a particular study (Triandini, Jayanatha, Indrawan, Werla Putra, & Iswara, 2019). Using this method, comparative reviews of automated software testing tools can be carried out in a systematic manner where the process follows the steps in the method. This study aims to provide a comparative review of some of the most discussed automated software testing tools from journals obtained by comparing the features of each software testing tool in terms of test types, software support, licenses, and so on using the SLR method. This is expected to assist software testers in selecting a software testing tool that fits their requirements.

**MATERIALS AND METHODS**

The method used in this study is the Systematic Literature Review (SLR). This method collects previous research related to the topic taken, then evaluates it, so as to reach a conclusion from the research objectives, namely to review the most discussed software automated testing tools from the journals obtained. This is done by comparing the features of each software testing tool in terms of test types, software support, licenses, and so on so as to assist software testers or Quality Assurance in selecting software automated testing tools that suit their requirements. This is in accordance with the objectives of SLR, namely through research on the topic of interesting phenomena with relevant research questions, then the research is identified, reviewed, evaluated, and interpreted. (Triandini et al., 2019). SLR consists of five stages: formulating problems, searching for literature, sorting literature search results, conducting analysis, and understanding the results of literature searches, as in Figure 1 (Izzati & Najwa, 2018)



Source: (Izzati & Najwa, 2018)

Figure 1. Methodology Literature Review

**Formulating Problems**

Based on the purpose of this study, namely reviewing software testing tools through reviews conducted on several studies, the first stage to take is to formulate the problem through several research questions (RQs). This formulation of the problem helps to gather all the information needed to analyze the different reviews. The research questions (RQs) in this study are:

1. *RQ1:* What are the automated software testing tools that are often discussed in journals?
2. *RQ2*: What are the characteristics of each of the automated software testing tools that are often discussed in journals?
3. *RQ3*: What are the advantages and disadvantages of each of the automated software testing tools that are often discussed in journals?

**Search for Literature**

The second stage in this method is to search for literature from journals related to automatic software testing tools to answer research questions (RQs) and research objectives. The search process is carried out using one of the addresses that are usually used to search scientific literature, https://scholar.google.com/. The first stage, searching for literature sources, 2 focus keywords used and associated with automated software testing tools, "Comparative of Automated Software Testing Tools" and "A Critical Analysis of Automated Software Testing Tools". This literature search only uses 2 main pages or 20 literature that first appears on Google Scholar according to each of the keywords used. From the results of this search, the next step is to sort out the results of the paper that match the predetermined criteria to get the paper that will be used in this study.

**Sorting Literature Search Results**

The third stage, namely sorting out the literature search results after searching using keywords that have been previously determined in the literature search section. Selection is done by determining if there are conditions that are not met in the criteria below, then automatically the paper cannot be used for research. The first selection criteria used were:

1. Elimination of paid papers.
2. Eliminate non-journal papers.
3. Eliminate papers whose publication years are not within the gap between 2010 - 2020 (the last 10 years).
4. Eliminate papers whose titles are not in English.
5. Eliminate papers with titles that are not related to automated software testing tools.

After identifying and analyzing the papers in the first selection, the results will be taken to the selection to analyze further papers, such as abstract analysis, keywords, and conclusions. How to make the second selection:

1. Eliminate papers with abstracts that are not related to comparisons, analyzes, case studies, and reviews of automated software testing tools.
2. Eliminate papers whose contents are not in English.
3. Eliminate papers that do not contain identification or publication statement.

After carrying out two stages of selection, the final stage of selecting search results is selecting duplicate papers. Based on all the selection stages, the number of papers used in the study was 11 papers. The paper will be used as the main reference in answering the problem formulation and producing conclusions to answer the objectives of this research.

**Perform Analysis**

The fourth stage of this method is to analyze the papers obtained in the third stage. The analysis was carried out by collecting and summarizing several literature search results related to automated software testing tools. The testing tools discussed in this paper were selected based on the 5 testing tools that were most widely discussed in the literature search results paper, then their strengths and weaknesses were analyzed.

**Understanding Search Results**

After the analysis is carried out, at this stage it is carried out to understand the analysis so that it can answer the research questions that have been formulated previously and answer the research objectives.

**RESULTS AND DISCUSSION**

Based on the results of literature searches and selection carried out on the collected papers, the total paper used in this discussion is 11 papers, of which 10 papers come from search results using the keyword "Comparative of Automated Software Testing Tools" and 1 paper from the keyword "A Critical Analysis of Automated Software Testing Tools ”. The five automated testing tools discussed are Selenium, QTP, Testcomplete, Worry and Ranorex.

**Selenium**

Selenium is one of the most popular Open Source automated testing tools (Monier & El-mahdy, 2015). Selenium consists of 4 basic components: Selenium IDE, Selenium RC, WebDriver, Selenium Grid (Malik & Gahlan, 2016). This tool provides a testing framework for web-based applications and supports a wide variety of framework types, as well as supports a variety of browsers, platforms and programming languages. The following are the results of the analysis conducted in this study regarding the strengths and weaknesses of Selenium.

The advantages of Selenium (Bharti & Dutt, 2014; Gamido & Gamido, 2019; Jagannatha et al., 2014; V. Jain & Rajnish, 2018; H. Kaur & Gupta, 2013; Singh & Tarika, 2014; Sualim et al., 2016):

1. Provides a free and open source Third Party AddOn / Extension / Plug-in for the Firefox web browser, so there are no licensing or renewal fees.
2. Provides a Graphical User Interface to record user actions using Firefox.
3. Easy to download from the internet using the Selenium website.
4. Multi-browser support (Chrome, Firefox, Opera, Internet Explorer, Safari) for the implementation of test cases.
5. Supports Pause Option.
6. Ease of recording when testing and supports record playback.
7. Supports access to control.
8. Able to export code to various programming languages.
9. Has an excellent code export feature.
10. Can check maximum error messages and provides multiple sources for data-driven testing.
11. Support code reuse.
12. Provide accessibility in ease of learning.
13. The test report is relatively easy to understand.
14. Provide history on previously running programs.
15. Selenium provides output in Graphical and Command Line.
16. Provide access to different external sources such as Excel, Csv, Xml.
17. The use of position coordinates during script recording is quite reliable.
18. Having a level of accuracy and completeness of web applications is very effective because it takes less time to solve errors.
19. The script can be executed in various browsers.
20. Supports scheduled execution with complex code.
21. Provide various types of records such as keywords, scripts, low level procedures.
22. Supports Report Generation (Html).
23. Supports Java, .Net, c #, Perl, PHP, Python, and Ruby.
24. Support PC / MAC / UNIX / Windows / LINUX platforms.
25. Used for web, desktop and mobile applications, but not for performance testing.

Disadvantages of Selenium (Bharti & Dutt, 2014; Gamido & Gamido, 2019; Jagannatha et al., 2014; V. Jain & Rajnish, 2018; H. Kaur & Gupta, 2013; Singh & Tarika, 2014):

1. A person must have programming skills and some usage skills to use Selenium.
2. Does not allow Command Insertion operations to be performed while running.
3. Manual / assistance is not provided.
4. No blunder taking care of and no information base testing
5. There is no office to show network impacts and no asset checking ability.
6. Does not support reporting of congenital defects.
7. Does not support file upload.

**QTP**

TP (Quick Test Professional) is a Windows-based programming testing tool used to test applications on the desktop or the web (Okezie et al., 2019). This device works by distinguishing objects in the UI of an application or page and playing out the ideal tasks (Monier & El-mahdy, 2015). QTP is the best application for Functional testing and Regression testing developed by Hewlett Packard (HP) (Malik & Gahlan, 2016). Following are the results of the analysis carried out in this study regarding the strengths and weaknesses of QTP.

The advantages of QTP (Bharti & Dutt, 2014; Gamido & Gamido, 2019; Imran et al., 2016; Jagannatha et al., 2014; A. Jain et al., 2014; H. Kaur & Gupta, 2013; M. Kaur & Kumari, 2011):

1. Permits to test Java applets and applications, and sight and sound articles in standard Windows applications and applications, Visual Basic 6 applications, and .NET framework applications.
2. Can computerize some "Non-UI" based Test Cases, for example, record framework activities and information base testing.
3. Easy to edit scripts, measure, navigate, rotate, parametrize and validate results.
4. Can work very well with database applications.
5. Effectively create the most exhaustive reports because of the accessibility of productive online assistance.
6. Provides three types of recording namely context-sensitive mode, analog mode and low level recording.
7. Provides facilities to record mouse movements, screen coordinates, keystrokes, and, objects and their properties.
8. Provide Auto Documentation facility.
9. Provide access to different external sources such as excel sheets, databases, text files, Xml.
10. Can separate scripts from data.
11. Has an inbuilt data table which provides an easier way for Data Driven Testing.
12. Provides a summary of each test step and a graphical representation of the results.
13. Allows reuse of scripts on new builds as well as different object repositories of name mappings.
14. Scripting Engine does not need to be installed exclusively because it is part of the Windows OS.
15. Supports graphical interface record playback.
16. Easy to learn.
17. Can use various kinds of IDE such as Eclipse, Netbeans, Visual Studio.
18. Test results can be converted to different formats.
19. Supports default defect reporting, all types of dialog boxes and all types of file uploads.
20. Contents must be created in VBScript or JavaScript. QTP can support other languages ​​(java, .net, Delphi) but via add-ins.
21. Supports Windows XP, Window 7, Window Vista, Window 8 / 8.1.
22. Provides three recording modes: Normal or context-sensitive mode, Low-level mode, Analog mode.
23. Provide various tests such as Regression, Functional, Unit, Distributed and Web testing.
24. Browsers that support Google Chrome, Internet Explorer, Firefox.
25. Supports Report Generation (Html, Xml).

Disadvantages of QTP (Bharti & Dutt, 2014; Gamido & Gamido, 2019; Imran et al., 2016; Jagannatha et al., 2014; A. Jain et al., 2014; H. Kaur & Gupta, 2013; M. Kaur & Kumari, 2011):

1. Licensed so it is paid and very expensive, as well as an annual maintenance fee.
2. Only the client server application and also supports add-on, but users need to purchase a license for it.
3. It is difficult to implement ASAP tests for web applications using QTP, especially with Windows 7.
4. Does not provide easy access to controls.
5. Does not support Pause Option in the middle of testing.
6. Does not support Insert Commands while recording.
7. Sometimes it doesn't recognize object during script playback.
8. Using more CPU and RAM.
9. Requires programming knowledge.
10. Does not support scheduled execution.
11. Provides only web service testing.

**Test Complete**

Test Complete is a completely useful robotized testing stage created by SmartBear Software (Malik & Gahlan, 2016). This tool provides recording and scripting for testing (Monier & El-mahdy, 2015). TestComplete allows testers to generate tests for Windows, Web, Android, and iOS applications and supports a wide variety of test types and methods, for example; functional testing, unit testing, and GUI testing (Okezie et al., 2019). TestComplete requires a license to be used after a free 30-day trial. The following are the results of the analysis conducted in this study regarding the strengths and weaknesses of TestComplete.

The advantages of TestComplete (Bharti & Dutt, 2014; Gamido & Gamido, 2019; H. Kaur & Gupta, 2013; M. Kaur & Kumari, 2011; Sualim et al., 2016):

1. There are no modules or additional items to purchase so you can introduce and make any test against any application.
2. Supports report generation and allows generating dump files.
3. Provides very easy access to controls.
4. Provide various types of records such as keywords, scripts.
5. Supports Pause Option in the middle of testing.
6. Provides facilities to record mouse movements, screen coordinates, keystrokes, and objects and their properties.
7. Provide Auto Documentation facility.
8. Provide access to different external sources like excel sheet, database, csv, sql.
9. Can separate scripts from data.
10. Allows script reuse on new builds.
11. Supports record playback.
12. Uses less CPU and RAM.
13. Can detect maximum errors and also allow to interact with the database.
14. Supports scripts in VBScript, JSScript, DelphiScript, C ++ Script and C # Script.
15. Supports Windows 7, Windows Vista, Windows Server 2008 or later operating systems.
16. Good for web and desktop based applications and supports mobile applications.
17. Support for all 32-bit and 64-bit windows applications.
18. Works well with database applications.
19. Provides various types of testing such as Functional, Regression, Unit, Distributed, Load, Web, and Manual Testing.
20. Supporting browsers, namely Chrome, Firefox, Opera, Internet Explorer.
21. Supports Report Generation (Html, Xml).

Disadvantages of TestComplete (Bharti & Dutt, 2014; Gamido & Gamido, 2019; H. Kaur & Gupta, 2013; M. Kaur & Kumari, 2011):

1. Licensed, so it pays.
2. Does not provide a summary of each test step and a graphical representation of the results.
3. Sometimes it doesn't recognize object during script playback.
4. Requires programming experience and skills.

**Watir**

Watir (Web Application Testing in Ruby) is an open source automated testing tool for Ruby Libraries released under the BSD license for testing various web-based applications. (Singh & Tarika, 2014). Using this tool requires programming skills in Ruby (Okezie et al., 2019). Following are the results of the analysis conducted in this study regarding the strengths and weaknesses of Watir.

The advantages of Watir (Gamido & Gamido, 2019; Singh & Tarika, 2014; Sualim et al., 2016):

1. Is a simple, flexible and open source application that is used for testing automation.
2. Support code reuse.
3. Supports record playback.
4. Easy to learn.
5. Provide access to different external sources such as Xml, Excel.
6. Can check for error messages and also allows to interact with the database.
7. The level of accuracy and completeness of web applications is very effective because it takes less time to resolve errors.
8. Can detect high maximum error rate.
9. Supports MAC / Windows / LINUX platforms.
10. Supports various browsers, such as Internet Explorer, Firefox, Google Chrome, Opera, Safari.
11. Supports Ruby, Java, C #, .net.
12. Supports Report Generation (Html, Xml).
13. Supports desktop, web, and mobile application testing.

Disadvantages of Watir (Singh & Tarika, 2014):

1. Does not allow Command Insertion operation to be performed while running.
2. Supports Pause Option with 3rd Party Plugins.
3. Does not support access to control.
4. Unable to export code to multiple languages.
5. Requires additional import path for the code export feature.

**Ranorex**

Ranorex is a graphical user interface automation framework used for testing desktop, web-based and mobile applications (Gamido & Gamido, 2019). Ranorex performs tests based on image detection and facilities for recording and playing back (Monier & El-mahdy, 2015). The scripts are written in pure .net code using C #, VB.net, and Python. Following are the results of the analysis carried out in this study regarding the strengths and weaknesses of Ranorex.

The advantages of Ranorex (Anjum et al., 2017; Gamido & Gamido, 2019; A. Jain et al., 2014):

1. Support codeless working and coding capabilities.
2. Perform platform compatibility testing.
3. Has the flexibility to facilitate maintenance and software updates.
4. Has the ability for software components to be added, modified, and removed easily without affecting the existing system.
5. Has the ability to add components easily.
6. Has the ability to make changes to error correction, supported by the defined interface, documentation, comments in the code.
7. Can run on multiple platforms, such as operating systems, browsers, etc.
8. Supports record playback.
9. Easy to learn.
10. Provide access to different external sources such as Csv, Excel, Sql.
11. Supports executive summary with graphs for faster and better defect comparisons across each process.
12. A license for training can be obtained at a very low cost.
13. Has 'click and go Function' feature to ensure reuse of test actions and various UI elements with technical skill level team.
14. A graphical user interface (GUI) automation framework used for testing desktop, web-based, and mobile applications.
15. Supports Windows platform.
16. Supports cross-browser testing (Chrome, Firefox, Opera, IE, Netscape, Safari).
17. Supports VB Script, .net, C ++, C #, python.
18. Supports Report Generation (Html).

Disadvantages of Ranorex (Gamido & Gamido, 2019; A. Jain et al., 2014):

* + - 1. A low consumer base resulting in unavailability of trained resources.

**CONCLUSION**

This study has conducted an analysis of several automated software testing tools using the Systematic Literature Review (SLR) method to collect related papers. Based on the results of the literature search and selection carried out on the collected papers, the final total of the papers used for analysis in this study were 11 papers related to automatic software testing tools, while for automatic software testing tools discussed in this study were testing tools that most discussed in literature search results papers. Based on the analysis that has been carried out in this study, analyzing the strengths and weaknesses of Selenium, Quick Test Professional (QTP), TestComplete, Web Application Testing in Ruby (Watir), and Ranorex, it can be concluded that there is no automated software testing tool that absolutely perfect. Every software automated testing tool has its own strengths and weaknesses. The features possessed by each software automatic testing tool are different so that the selection of this testing tool must be based on the needs of the test to be carried out. The results of this study are expected to be used as a reference to assist software testers in choosing an automatic software testing tool that suits their testing needs. Suggestions that can be given for further research, namely using more references in the paper collection stage, so that the total final paper used for analysis is more and comparing with other automated testing tools, as well as conducting some direct testing using test equipment to adjust to developments the latest version of the test tool.

**REFERENCE**

Anjum, H., Imran, M., Jehanzeb, M., Khan, M., Chaudhry, S., Sultana, S., Shahid, Z., Zeshan, F., & Nazir, S. (2017). A Comparative Analysis of Quality Assurance of Mobile Applications using Automated Testing Tools. *International Journal of Advanced Computer Science and Applications*, *8*(7), 249–255. https://doi.org/10.14569/ijacsa.2017.080733

Bharti, T., & Dutt, E. V. (2014). *Relative Review of Automated Testing Tools : ( QTP ) Quick Test Professional , Selenium and Test Complete testing is the process of giving the input to the*. *2*(6).

Gamido, H. V., & Gamido, M. V. (2019). Comparative review of the features of automated software testing tools. *International Journal of Electrical and Computer Engineering*, *9*(5), 4473–4478. https://doi.org/10.11591/ijece.v9i5.pp4473-4478

Imran, M., Hebaishy, M. A., & Alotaibi, A. S. (2016). A Comparative Study of QTP and Load Runner Automated Testing Tools and their Contributions to Software Project Scenario. *International Journal of Innovative Research in Computer and Communication Engineering*, *4*(1), 457–466. https://doi.org/10.15680/IJIRCCE.2016

Izzati, A. N., & Najwa, N. F. (2018). Pengaruh Stakeholder Perspective Dalam Penerapan ERP: A Systematic Literature Review. *Jurnal Teknologi Informasi Dan Ilmu Komputer*, *5*(1), 41. https://doi.org/10.25126/jtiik.201851540

Jagannatha, S., Niranjanamurthy, M., Manushree, S., & Chaitra, G. (2014). Comparative Study on Automation Testing using Selenium Testing Framework and QTP. *International Journal of Computer Science and Mobile Computing*, *310*(10), 258–267.

Jain, A., Jain, M., & Dhankar, S. (2014). A Comparison of RANOREX and QTP Automated Testing Tools and their impact on Software Testing. *Ijems*, *1*(1), 8–12.

Jain, V., & Rajnish, K. (2018). Comparative Study of Software Automation Testing Tools : OpenScript and Selenium. *Journal of Engineering Research and Application*, *8*(2), 29–33. https://doi.org/10.9790/9622-0802032933

Kaur, H., & Gupta, G. (2013). Comparative Study of Automated Testing Tools: Selenium, Quick Test Professional and Testcomplete. *International Journal of Engineering Research and Applications*, *3*(5), 1739–1743. https://doi.org/10.18701/imsmanthan.v11i01.6879

Kaur, M., & Kumari, R. (2011). Comparative Study of Automated Testing Tools: TestComplete and QuickTest Pro. *International Journal of Computer Applications*, *24*(1), 1–7. https://doi.org/10.5120/2918-3844

Mahmood, H., & Sirshar, M. (2017). A Case Study of Web Based Application by Analyzing Performance of a Testing Tool. *International Journal of Education and Management Engineering*, *7*(4), 51–58. https://doi.org/10.5815/ijeme.2017.04.06

Malik, V., & Gahlan, M. (2016). Comparative Study of Automated Web Testing Tools. *International Journal of Latest Trends in Engineering and Technology*, *6*(3), 8. http://www.ijltet.org/journal/55.pdf

Malve, S., & Sharma, P. (2017). Investigation of Manual and Automation Testing using Assorted Approaches. *International Journal of Scientific Research in Computer Science and Engineering*, *5*(2), 81–87.

Monier, M., & El-mahdy, M. M. (2015). Evaluation of automated web testing tools. *International Journal of Computer Applications Technology and Research*, *4*(5), 405–408. https://doi.org/10.7753/ijcatr0405.1014

Okezie, F., Odun-Ayo, I., & Bogle, S. (2019). A Critical Analysis of Software Testing Tools. *Journal of Physics: Conference Series*, *1378*(4). https://doi.org/10.1088/1742-6596/1378/4/042030

Shalini, & Thakur, J. (2017). Web Performance Testing Tools – A Review. *International Journal for Research in Applied Science and Engineering Technology*, *V*(X), 1057–1063. https://doi.org/10.22214/ijraset.2017.10154

Shaukat, K., Shaukat, U., Feroz, F., Kayani, S., & Akbar, A. (2015). Taxonomy of Automated Software Testing Tools. *International Journal of Computer Science and Innovation*, *2015*(1), 7–18. http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.735.4191&rep=rep1&type=pdf

Singh, I., & Tarika, B. (2014). Comparative Analysis of Open Source Automated Software Testing Tools: Selenium, Sikuli and Watir. *International Journal of Information & Computation Technology*, *4*(15), 1507–1518. https://doi.org/10.13140/2.1.1418.4324

Sualim, S. A., Yassin, N. M., & Mohamad, R. (2016). Comparative Evaluation of Automated User Acceptance Testing Tool for Web Based Application. *International Journal of Software Engineering and Technology 02(2) 1-6 International*, *02*(2), 1–6. http://files/590/Sualim et al. - 2016 - Comparative Evaluation of Automated User Acceptanc.pdf

Triandini, E., Jayanatha, S., Indrawan, A., Werla Putra, G., & Iswara, B. (2019). Metode Systematic Literature Review untuk Identifikasi Platform dan Metode Pengembangan Sistem Informasi di Indonesia. *Indonesian Journal of Information Systems*, *1*(2), 63. https://doi.org/10.24002/ijis.v1i2.1916