ABDI TEKNOYASA

ISSSN: 2745-701X
E-ISSN: 2745-7028
http://journals2.ums.ac.id/index.php/abditeknoyasa/

DEVELOPMENT OF QUALITY ASSURANCE AUTOMATIC TESTING SCRIPT TO INCREASE TESTING EFFICIENCY FOR MOBILE APPLICATIONS

Abdi Suryadinata Telaga *
Construction and Building Engineering Department
Politeknik Astra Komp. Astra International Tbk
abdi.telaga@polman.astra.ac.id

Lita Eka Wulansari
Department System Information,
Politeknik Astra. Komp. Astra
International Tbk
litaekawulansari@gmail.com

Naufal Norman Hisyam
Construction and Building Engineering Department
Politeknik Astra Komp. Astra International Tbk
naufal.norman.hisyam@gmail.com

***** Corresponding author

Naskah dikirim 9 Desember 2022
Naskah direvisi 14 Desember 2022
Naskah diterima 14 Desember 2022

ABSTRACT

Consumers increasingly use mobile applications because of the convenience and mobility of use. However, the short application development time and demands to maintain application quality and user satisfaction require a fast and thorough testing process. Many software testers currently test their software manually, causing problems, such as longer test times, inconsistent testing, and the need for human intervention to perform all tests. Therefore, a Quality Assurance Automatic Testing (QA AT) system was created based on mind map diagrams to overcome the earlier problems. QA AT is a script designed to verify the software requirements. The method to develop the QA AT script is as follow. Firstly, develop a Business Process diagram. Secondly, create the Mind Map diagram. Thirdly, build test cases scenarios based on the mind map diagram. Then translate the testing scenarios into testing scripts. Finally, the test scripts run on Katalon Studio platform. The QA AT system transforms the testing process into automated testing. The results show that automated testing can shorten test time by 86%,. Further, reducing human involvement, creating a consistent testing environment, and providing a detailed report.

KEYWORDS: Development, Quality Assurance, Mobile Application

INTRODUCTION

Currently, the price of smartphones is increasingly affordable for all people. The increasing use of smartphones makes the demand for mobile applications increase. This increase is due to ease of use and mobility. However, technological advances, tight competition and customer demand for new features in mobile applications have made the development period of mobile applications abridged. Meanwhile, the quality of the mobile application is one of the factors that determine user acceptance of the application.

Therefore, the application testing phase is crucial to ensure that the application meets user expectations. Testing a mobile app is very different from testing a desktop app. Mobile applications development poses many challenges, such as a short application development period, variations in device screen sizes, and varying operating systems. Meanwhile, the testing phase is the most time-consuming phase of software development. The testing phase is estimated to take about 60% of the overall software development time. In addition, testing mobile applications is also very tedious because the testers must be able to guess the user's behaviour when using the mobile application.

Furthermore, the testers also must repeat the user's behaviour so that the test can meet the requirements. Therefore, the need to automate the test becomes very urgent. Particularly in agile software development, testing automation is critical to overcoming short application development time (Collins & Lucena, 2012). Therefore, the research tried to answer two questions. Firstly, what is the suitable methodology to capture user behaviour and map the behaviour into testing scenario variation? Secondly, how much time can be saved by automated testing from manual testing?.

STUDY LITERATURE

Testing mobile applications is a meticulous job. Software testers must understand the application user's behaviour and understand the mobile application itself. The testers often try to understand the application through the applications' graphical user interface (GUI) and the code provided by the developer. However, in some development companies, there is a separation between the application development team and the quality assurance (QA) team. Therefore, the QA team often does not get access to the mobile application code. Therefore, testing must be done black box.

Therefore, with black-box testing in mind to complement automated testing, researchers use a visual perspective to aid testers in building testing scenarios (Xue, 2021). Similarly, Li et al. (2015) used user-guided automation to test mobile applications (Li et al., 2014). Furthermore, Farto and Endo (2015) used a model-based testing approach to aid testers in creating testing scenarios (de Cleva Farto & Endo, 2015). However, many software development companies in Indonesia are unfamiliar with that methodology and find the method is too complex in practical use. Therefore, the testers must find a methodology that is able to provide visual information and is easy to understand.

The Mind Map diagram is a diagram that visually organizes information. The diagram allows complex information to be presented in a single concept. Therefore, mind map diagrams are easier to explain information visually to help tester manage information in building testing scenarios. Software testers have used the Mind Map to develop the testing scenario. Otaduy and Diaz (2017) combine a wiki-based and the Mind Map diagram to create user acceptance test scenarios (Otaduy & Diaz, 2017). Similarly, Rajasekaran and Nithyarao (2017) utilize The Mind Map diagram to improve the efficiency of software testing (Rajasekaran & Nithyarao, 2017). The research found that the Mind Map is an efficient method to create lean testing scenarios. Furthermore, the method is also adaptive to user requirement changes. Therefore, saving time in the software testing phase. For that reason, the study used the Mind Map diagram to capture mobile app user behaviour visually then develop test cases based on the information.

METHOD

The development of the QA AT script on this mobile application follows Scrum methodology. The software used in managing the QA AT system in this application is Jira software. Whereas, Katalon studio is used as automation testing software. Because the tester does not know any code behind the mobile app then the testing method is black-box testing. The testing is performed in a development environment because testing in the operational environment cannot be carried out due to the application has not been deployed. However, the data obtained should not be much different from the operating environment

The methodology for software testing is as follow:

1. Develop Business Process Diagram to understand the purpose of the mobile app.
2. Develop Mind Map diagrams to depict users' minds when using the mobile application visually.

3. Develop test cases based on the Mind Map diagram.
4. Script creation based on mind map diagram.
5. Perform automation regression test by running testing script in Katalon Studio. The script is done through a test suite that generates a test result report based on the script.
6. Calculate reduction time. The reduction test result is calculated based on the formula below.

$$P=100\%-(X2/X1 \ \times 100\%) \qquad (1)$$

Where,
P = Reduction time
X2 = Automated testing time
X1 = Manual testing time

RESULTS

System testing is carried out to ensure that the system meets user requirements and can be used in a production environment. Furthermore, the testing is also necessary to confirm that no errors or bugs in the system. QA AT testing is performed by making a test suite for regression testing. Regression testing is the test that includes all test cases or scripts that have been created. Therefore, the test suite that is run in this test is the entire test case that has been put together.

The data from the current test results are obtained from the development environment because testing in the operational environment cannot be carried out because the application deployment has not been implemented. Table 1 shows the testing result, whereas Table 2 shows testing results from rerun false-positive results.

Table 1. Run Test Suites Detail

No	Name	Testing Type	Execution Time
1	Run 1	Regression Test	1 hour 23 minutes
2	Run 2	Regression Test	1 hour 16 minutes
3	Run 3	Regression Test	1 hour 24 minutes
4	Run 4	Regression Test	1 hour 15 minutes
5	Run 5	Regression Test	1 hour 20 minutes
6	Run 6	Regression Test	1 hour 17 minutes
7	Run 7	Regression Test	1 hour 20 minutes
8	Run 8	Regression Test	1 hour 21 minutes

Table 2. Rerun Test Suites Detail

No	Name	Testing Type	Execution Time
1	Run 1	Rerun Regression Test	5 minutes
2	Run 2	Rerun Regression Test	7 minutes
3	Run 3	Rerun Regression Test	5 minutes
4	Run 4	Rerun Regression Test	8 minutes
5	Run 5	Rerun Regression Test	5 minutes
6	Run 6	Rerun Regression Test	7 minutes
7	Run 7	Rerun Regression Test	6 minutes
8	Run 8	Rerun Regression Test	6 minutes

DISCUSSION

System testing is carried out to ensure that in the application testing stage, consistent data or input is needed to get consistent results in each test. In manual testing, the input used in test cases may differ between test sessions or between testers. Therefore, there is a possibility of variation in test results between test sessions or between testers. Whereas, in the QA AT system, all inputs that are used in the test cases are obtained in the form of variables that always be similar every time they are tested. Therefore, the testing process guarantees that the test results always be the same and stable on all tests.

Mobile application testing that is performed manually takes approximately ten working hours or more than one day. In contrast, application testing with the QA AT script that has been created takes an average of 1 hour 20 minutes. The data obtained from the test sample can be seen in detail in Table 1. Besides the automated testing time, additional time is required to re-execute the test cases listed as failing. Further testing is necessary to ensure that the automated tests that have been carried out do not have false positives anymore. The process takes an average of 6 minutes which can be seen in detail in Table 2.

The average time required to perform tests automatically using the QA AT test is 1 hour, 26 minutes or 86 minutes. Therefore, using the formula above, the percentage of time reduction in testing using the QA AT system from manual testing is 86%. The result shows that the Mind Map diagram can aid testers to develop test scenarios based on user behaviour. The test scenario is translated to the testing script used in Katalon Studio to automate the testing process to reduce testing time.

CONCLUSION

Mind map diagram can aid tester to understand user behaviour. Therefore, it helps testers to build testing scenarios. The QA AT system on testing mobile applications can shorten the testing time by reducing the test time by 86% from the manual testing time. Furthermore, the application of QA AT can improve the consistency of application testing. With QA AT, all testing procedures follow similar orders and have consistent data. Therefore, reducing human involvement, creating a consistent testing environment, and providing a detailed report.

REFERENCES

[1] Collins, E. F., & Lucena, V. F. de. (2012). Software Test Automation practices in agile development environment: An industry experience report. *2012 7th International Workshop on Automation of Software Test (AST)*, 57–63. https://doi.org/10.1109/IWAST.2012.6228991

[2] de Cleva Farto, G., & Endo, A. T. (2015). Evaluating the Model-Based Testing Approach in the Context of Mobile Applications. *Electronic Notes in Theoretical Computer Science*, *314*, 3–21. https://doi.org/https://doi.org/10.1016/j.entcs.2015.05.002.

[3] Li, X., Jiang, Y., Liu, Y., Xu, C., Ma, X., & Lu, J. (2014). User Guided Automation for Testing Mobile Apps. *2014 21st Asia-Pacific Software Engineering Conference*, *1*, 27–34. https://doi.org/10.1109/APSEC.2014.13

[4] Otaduy, I., & Diaz, O. (2017). User acceptance testing for Agile-developed web-based applications: Empowering customers through wikis and mind maps. *Journal of Systems and Software*, *133*, 212–229. https://doi.org/https://doi.org/10.1016/j.jss.2017.01.002

[5] Rajasekaran, K., & Nithyarao, T. K. (2017). The Effective New Frame Work for Mind Mapping Matrix Test Case Techniques. *International Journal of Innovative Science and Research Technology*, *2*(7), 470–474.

[6] Xue, F. (2021). Automated mobile apps testing from visual perspective. *ISSTA 2020: Proceedings of the 29th ACM SIGSOFT International Symposium on Software Testing and Analysis*, 577–581. https://doi.org/10.1145/3395363.3402644.

Plagiarism Checker X - Report

Originality Assessment

Overall Similarity: **14%**

Date: Dec 13, 2021

Statistics: 255 words Plagiarized / 1863 Total words

Remarks: Low similarity detected, check your supervisor if changes are required.

Development of Quality Assurance Automatic Testing Script to Increase Testing Efficiency for Mobile Applications Abstract. Consumers increasingly use mobile applications because of the convenience and mobility of use. However, the short application development time and demands to maintain application quality and user satisfaction require a fast and thorough testing process. Many software testers currently test their software manually, causing problems, such as longer test times, inconsistent testing, and [1]the need for human intervention to perform all tests. Therefore, a Quality Assurance Automatic Testing (QA AT) system was created based on mind map diagrams to overcome the earlier problems. QA AT is a script designed to verify the software requirements. The method to develop the QA AT script is as follow. Firstly, develop a Business Process diagram. Secondly, create the Mind Map diagram. Thirdly, build test cases scenarios based on the mind map diagram. Then translate the testing scenarios into testing scripts. Finally, the test scripts run on Katalon Studio platform. The QA AT system transforms the testing process into automated testing. The results show that automated testing can shorten test time by 86%,. Further, reducing human involvement, creating a consistent testing environment, and providing a detailed report. 1. Introduction Currently, the price of smartphones is increasingly affordable for all people. The increasing use of smartphones makes the demand [1]for mobile applications increase. This increase is due to ease of use and mobility. However, technological advances, tight competition and customer demand for new features in mobile applications have made the development period of mobile applications abridged. Meanwhile, [1]the quality of the mobile application is one of the factors that determine user acceptance of the application. Therefore, the application testing phase is crucial to ensure that the application meets user expectations. Testing a mobile app is very different from testing a desktop app. Mobile applications development poses many challenges, such as a short application development period, variations in device screen sizes, and varying operating systems. Meanwhile, the testing phase is the most time-consuming phase of software development. The testing phase is estimated to take about 60% of the overall software development time. In addition, testing mobile applications is also very tedious because the testers must

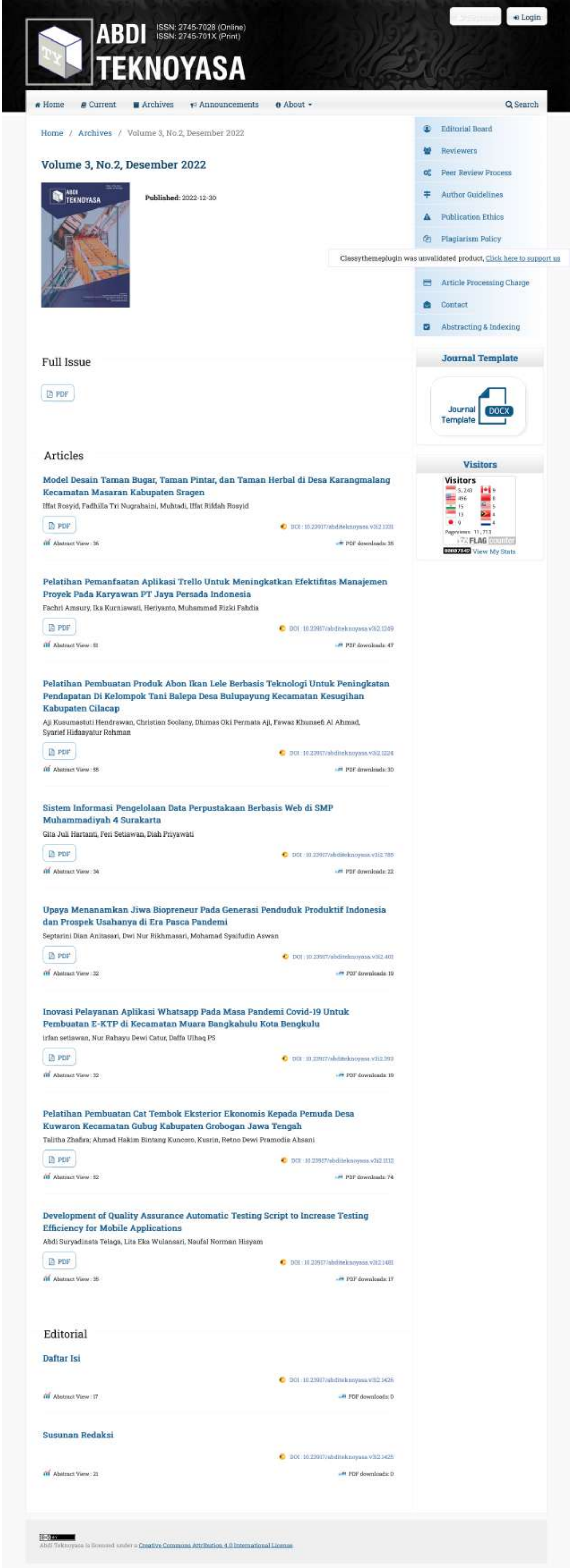
[1]be able to guess the user's behaviour when using the mobile application. Furthermore, the testers also must repeat the user's behaviour so that the test can meet the requirements. Therefore, [2]the need to automate the test becomes very urgent. Particularly in agile software development, testing automation [1]is critical to overcoming short application development time (1). Therefore, the research tried to answer two questions. Firstly, what is the suitable methodology to capture user behaviour and map the behaviour into testing scenario variation? Secondly, how much time can be saved by automated testing from manual testing?   2. Study literature Testing mobile applications is a meticulous job. Software testers must understand the application user's behaviour and understand the mobile application itself. The testers often try to understand the application through the applications' graphical user interface (GUI) and the code provided by the developer. However, in some development companies, there is a separation between the application development team and the quality assurance (QA) team. Therefore, the QA team often does not get access to the mobile application code. Therefore, testing must be done black box.   Therefore, with black-box testing in mind to complement automated testing, researchers use a visual perspective to aid testers [2]in building testing scenarios (2). Similarly, Li et al. (2015) used user-guided automation to test mobile applications (3). Furthermore, Farto and Endo (2015) used a model-based testing approach to aid testers in creating testing scenarios (4). However, many software development companies in Indonesia are unfamiliar with that methodology and find [1]the method is too complex in practical use. Therefore, the testers must find a methodology [3]that is able to provide visual information and is easy to understand.   The Mind Map diagram is a diagram that visually organizes information. The diagram allows complex information to be presented in a single concept.   Therefore, mind map diagrams are easier to explain information visually to help tester manage information [2]in building testing scenarios. Software testers have used the Mind Map to develop the testing scenario. Otaduy and Diaz (2017) combine a wiki-based and the Mind Map diagram to create user acceptance test scenarios (5). Similarly, Rajasekaran and Nithyarao (2017) utilize The Mind Map diagram [1]to improve the efficiency

of software testing (6). The research found that the Mind Map is an efficient method to create lean testing scenarios. Furthermore, the method is also adaptive to user requirement changes. Therefore, saving time in the software testing phase. For that reason, the study used the Mind Map diagram to capture mobile app user behaviour visually then develop test cases [3]based on the information. 3. METHODOLOGY The development of the QA AT script on this mobile application follows Scrum methodology. The software used in managing the QA AT system in this application is Jira software. Whereas, Katalon studio is used as automation testing software. Because the tester does not know any code behind the mobile app [4]then the testing method is black-box testing. The [2]testing is performed in a development environment because testing in the operational environment cannot be carried out due to the application has not been deployed. However, the data obtained should not be much different from the operating environment The methodology [3]for software testing is as follow: 1. Develop Business Process Diagram to understand the purpose of the mobile app. 2. Develop Mind Map diagrams to depict users' minds [1]when using the mobile application visually. 3. Develop test cases based on the Mind Map diagram. 4. Script creation based on mind map diagram. 5. Perform automation regression test by running testing script in Katalon Studio. The script is done through [3]a test suite that generates a test result report based on the script. 6. Calculate reduction time. The reduction test result is calculated [1]based on the formula below. P=100%-(X2/X1 x 100%) (1) Where, P = Reduction time X2 = Automated testing time X1 = Manual testing time 4. RESULTS System [4]testing is carried out to ensure that the system meets user requirements and can be used in a production environment. Furthermore, the testing is also necessary to confirm that no errors or bugs in the system. QA AT [1]testing is performed by making a test suite for regression testing. Regression testing is the test that includes all test cases or scripts that have been created. Therefore, the test suite that is run in this test is the entire test case that [2]has been put together. The data from the current test results are obtained from the development environment because testing in the operational environment cannot be carried out because the application deployment has not been implemented. Table 1 shows the testing result, whereas Table 2 shows testing results from rerun false-positive results. Table 1. Run Test Suites Detail No Name Testing Type Execution Time 1 Run 1 Regression Test 1 hour 23 minutes 2 Run 2 Regression Test 1 hour 16 minutes 3 Run 3 Regression Test 1 hour 24 minutes 4 Run 4 Regression Test 1 hour 15 minutes 5 Run 5 Regression Test 1 hour 20 minutes 6 Run 6 Regression Test 1 hour 17 minutes 7 Run 7 Regression Test 1 hour 20 minutes 8 Run 8 Regression Test 1 hour 21 minutes Table 2. Rerun Test Suites Detail No Name Testing Type Execution Time 1 Run 1 Rerun Regression Test 5 minutes 2 Run 2 Rerun Regression Test 7 minutes 3 Run 3 Rerun Regression Test 5 minutes 4 Run 4 Rerun Regression Test 8 minutes 5 Run 5 Rerun Regression Test 5 minutes 6 Run 6 Rerun Regression Test 7 minutes 7 Run 7 Rerun Regression Test 6 minutes 8 Run 8 Rerun Regression Test 6 minutes 5. DISCUSSION System [4]testing is carried out to ensure that in the application testing stage, consistent data or input is needed to get consistent results in each test. In manual testing, the input used in test cases may differ between test sessions or between testers. Therefore, [1]there is a possibility of variation in test results between test sessions or between testers. Whereas, in the QA AT system, all inputs that are used in the test cases are obtained in the form of variables that always be similar every time they are tested. Therefore, [2]the testing process guarantees that the test results always be the same and stable on all tests.

Mobile application testing that is performed manually takes approximately ten working hours or more than one day. In contrast, application testing with the QA AT script that has been created takes an average of 1 hour 20 minutes. The data obtained from the test sample can be seen in detail in Table 1. Besides the automated testing time, additional time is required to re-execute the test cases listed as failing. Further testing is necessary to ensure that the automated tests that have been carried out do not have false positives anymore. The process takes [1]an average of 6 minutes which can be seen in detail in Table 2. The average time required to perform tests automatically using the QA AT test is 1 hour, 26 minutes or 86 minutes. Therefore, using the formula above, the percentage of time reduction in testing using the QA AT system [3]from manual testing is 86%. The result shows that the Mind Map diagram can aid testers to develop test scenarios based on user behaviour. The test scenario is translated to the testing script used in Katalon Studio to automate the testing process to reduce testing time.   6. Conclusion Mind map diagram can aid tester to understand user behaviour. Therefore, it helps testers to build testing scenarios. The QA AT system on testing mobile applications can shorten the testing time by reducing the test time by 86% from the manual testing time. Furthermore, [1]the application of QA AT can improve the consistency of application testing. With QA AT, all testing procedures follow similar orders and have consistent data. Therefore, reducing human involvement, creating a consistent testing environment, and providing a detailed report. References 1. Collins EF, Lucena VF de. Software Test Automation practices in agile development environment: An industry experience report. In: 2012 7th [5]International Workshop on Automation of Software Test (AST). 2012. p. 57–63. 2. Xue F. Automated mobile apps testing from visual perspective. In: ISSTA 2020: [2]Proceedings of the 29th ACM SIGSOFT International Symposium on Software Testing and Analysis [Internet]. ACM; 2021. p. 577–581. Available from: https://dl.acm.org/doi/10.1145/3395363.3402644#sec-ref 3.   Li X, Jiang Y, Liu Y, Xu C, Ma X, Lu J. User Guided Automation [3]for Testing Mobile Apps. In: 2014 21st Asia-Pacific Software Engineering Conference. 2014. p. 27–34. 4. de Cleva Farto G, Endo AT. Evaluating the Model-Based Testing Approach in the Context of Mobile Applications. Electron Notes Theor Comput Sci [Internet]. 2015;314:3–21. Available from: https://www.sciencedirect.com/science/article/pii/S1571066115000250 5.   Otaduy I, Diaz O. [4]User acceptance testing for Agile-developed web-based applications: Empowering customers through wikis and mind maps. J Syst Softw [Internet]. 2017;133:212–29. Available from: https://www.sciencedirect.com/science/article/pii/S016412121730002X 6. Rajasekaran K, Nithyarao TK. The Effective New Frame Work for Mind Mapping Matrix Test Case Techniques. Int J Innov Sci Res Technol. 2017;2(7):470–4.

Sources

1	https://thesai.org/Publications/ViewIssue?volume=8&issue=1&code=IJACSA INTERNET 5%
2	https://www.researchgate.net/publication/266144501_Risk_Factors_in_Software_Development_Phases INTERNET 3%
3	https://www.researchgate.net/publication/301691086_MobiPlay_A_Remote_Execution_Based_Record-and-Replay_Tool_for_Mobile_Applications INTERNET 3%
4	https://www.softwaretestinghelp.com/types-of-software-testing/ INTERNET 2%
5	https://www.tugraz.at/institutes/ist/publications/all-publications/ INTERNET <1%



ABDI TEKNOYASA
ISSN: 2745-7028 (Online)
ISSN: 2745-701X (Print)

Home / Archives / Volume 3, No.2, Desember 2022

Volume 3, No.2, Desember 2022

Published: 2022-12-30

Classythemeplugin was unvalidated product, Click here to support us

Full Issue

PDF

Articles

Model Desain Taman Bugar, Taman Pintar, dan Taman Herbal di Desa Karangmalang Kecamatan Masaran Kabupaten Sragen
Iffat Rosyid, Fadhilla Tri Nugrahaini, Muhtadi, Iffat Rifdah Rosyid
PDF DOI : 10.23917/abditeknoyasa.v3i2.1331
Abstract View : 36 PDF downloads: 35

Pelatihan Pemanfaatan Aplikasi Trello Untuk Meningkatkan Efektifitas Manajemen Proyek Pada Karyawan PT Jaya Persada Indonesia
Fachri Amsury, Ika Kurniawati, Heriyanto, Muhammad Rizki Fahdia
PDF DOI : 10.23917/abditeknoyasa.v3i2.1249
Abstract View : 51 PDF downloads: 47

Pelatihan Pembuatan Produk Abon Ikan Lele Berbasis Teknologi Untuk Peningkatan Pendapatan Di Kelompok Tani Balepa Desa Bulupayung Kecamatan Kesugihan Kabupaten Cilacap
Aji Kusumastuti Hendrawan, Christian Soolany, Dhimas Oki Permata Aji, Fawaz Khunaefi Al Ahmad, Syarief Hidaayatur Rohman
PDF DOI : 10.23917/abditeknoyasa.v3i2.1224
Abstract View : 55 PDF downloads: 30

Sistem Informasi Pengelolaan Data Perpustakaan Berbasis Web di SMP Muhammadiyah 4 Surakarta
Gita Juli Hartanti, Feri Setiawan, Diah Priyawati
PDF DOI : 10.23917/abditeknoyasa.v3i2.785
Abstract View : 34 PDF downloads: 22

Upaya Menanamkan Jiwa Biopreneur Pada Generasi Penduduk Produktif Indonesia dan Prospek Usahanya di Era Pasca Pandemi
Septarini Dian Anitasari, Dwi Nur Rikhmasari, Mohamad Syaifudin Aswan
PDF DOI : 10.23917/abditeknoyasa.v3i2.401
Abstract View : 32 PDF downloads: 19

Inovasi Pelayanan Aplikasi Whatsapp Pada Masa Pandemi Covid-19 Untuk Pembuatan E-KTP di Kecamatan Muara Bangkahulu Kota Bengkulu
irfan setiawan, Nur Rahayu Dewi Catur, Daffa Ulhaq PS
PDF DOI : 10.23917/abditeknoyasa.v3i2.393
Abstract View : 32 PDF downloads: 19

Pelatihan Pembuatan Cat Tembok Eksterior Ekonomis Kepada Pemuda Desa Kuwaron Kecamatan Gubug Kabupaten Grobogan Jawa Tengah
Talitha Zhafira; Ahmad Hakim Bintang Kuncoro, Kusrin, Retno Dewi Pramodia Ahsani
PDF DOI : 10.23917/abditeknoyasa.v3i2.1112
Abstract View : 52 PDF downloads: 74

Development of Quality Assurance Automatic Testing Script to Increase Testing Efficiency for Mobile Applications
Abdi Suryadinata Telaga, Lita Eka Wulansari, Naufal Norman Hisyam
PDF DOI : 10.23917/abditeknoyasa.v3i2.1481
Abstract View : 35 PDF downloads: 17

Editorial

Daftar Isi
DOI : 10.23917/abditeknoyasa.v3i2.1426
Abstract View : 17 PDF downloads: 0

Susunan Redaksi
DOI : 10.23917/abditeknoyasa.v3i2.1425
Abstract View : 21 PDF downloads: 0

Journal Template

Visitors