Jurnal Teknik Informatika (JUTIF)
Vol. 5, No. 4, August 2024, pp. 283-289

DOI: https://doi.org/10.52436/1.jutif.2024.5.4.2158
p-ISSN: 2723-3863
e-ISSN: 2723-3871

EVALUATION EXECUTION TIME FEATURES OF SIMPATI WEB-BASED MONITORING AND EVALUATION APPLICATION USING AUTOMATION TESTING

Nur Arifin[1], Ninik Agustin[*2], Tri Anggoro[3]

[1,2,3]Informatics, Mathematics and Computer Science Faculty, Universitas Nahdlatul Ulama Al Ghazali, Indonesia
Email: [1]arifin.202221026@students.unugha.id, [2]ninik.agustin@unugha.id, [3]trianggoro1103@gmail.com

(Article received: May 05, 2024; Revision: June 19, 2024; published: August 02, 2024)

Abstract

One of the data collection methods employed in monitoring and evaluation is a survey. The advantages of information technology-based surveys include a reduced risk of data loss and the ability to access the survey system from any location. Consequently, it is imperative to devise a web-based monitoring and evaluation system. In this study, the monitoring and evaluation system was developed using an Agile methodology and implementing automation testing. The Agile development method was utilized to create this application through the stages of design, development, testing, deployment, and feedback. The system was developed using the PHP programming language and the MySQL database. Automation testing was conducted using Katalon Studio. The design and implementation of the web-based SIMPATI application resulted in a system capable of monitoring and evaluating higher education with main features such as multi-user support, online surveys, a real-time database, and data visualization. Automation testing using Katalon Studio demonstrated that all features in the system run well and are stable. However, there is a discrepancy in the execution time for each test case. This discrepancy is attributed to the intricacy of the test cases and external factors such as system load, server performance, and network conditions. The fastest execution time for the login feature is 11 seconds, while the longest execution time for the add new user feature is 4 minutes and 29 seconds. In conclusion, automation testing in web-based system development employing agile methods can assist in the rapid and repeatable evaluation of system performance.

Keywords: *Automation Testing, Execution Time Features, Katalon Studio, Monitoring and Evaluation Systems*.

1. INTRODUCTION

The collection of data represents a pivotal phase in the research process, as well as in the decision-making process within organizational contexts. The objective of data collection is to enable the evaluation and analysis of companies, organizations, and institutions. This can be achieved through the administration of surveys, which can be conducted manually or online via a survey system. The former approach, namely the distribution of paper survey forms to respondents who will complete the survey, is undoubtedly time-consuming. In contrast, online surveys permit respondents to complete the survey via a website-based survey system or mobile application [1][2].

The advantages of web-based surveys are numerous and include ease of use, cost savings, reduced risk of data loss, and the ability to access the survey system from any location. The web-based survey system can be accessed from any internet-enabled device, allowing respondents to complete surveys at their convenience. Once a survey is completed, the data is stored directly in the database, eliminating the need for manual data entry and reducing the time required to process the data. The results can be displayed in real-time, providing a more efficient and timely approach to data analysis [2]. One method of implementing a website-based survey system is through the use of a monitoring and evaluation system.

Web-based monitoring and evaluation systems can collect and observe data, enhance the efficacy of a program or project, and minimize the occurrence of human error in comparison to manual monitoring and evaluation [3]. Monitoring itself is an activity that is carried out continuously in stages to progress a program or work by the plan, and identify problems that arise to be resolved immediately [4]. The monitoring process can be carried out by collecting the necessary information data [5][6]. The utilization of the data collected during or after program implementation will produce outputs that can be seen as key components in making decisions through assessment, known as evaluation [3].

Consequently, one means of enhancing the efficacy of higher education is through the implementation of monitoring and evaluation. The execution of manual monitoring entails the direct observation of performance, whether through in-person visits or meetings with the relevant parties, including students, lecturers, employees, collaborative partners, and alumni. This approach, however, necessitates a significant investment of time and energy. Consequently, it is imperative to develop a system-based monitoring and evaluation framework

to enhance the effectiveness and usability of the monitoring process [7].

One of the system development methodologies is the Agile Methodology. This methodology is a short-term software development methodology that necessitates the capacity for rapid adaptation to changes that occur during the development process to the user's needs and continuous software planning [8][9][10]. The characteristics of the agile methodology previously outlined indicate that it can be effectively employed in the development of monitoring and evaluation systems that necessitate a high degree of adaptation and communication with clients. The monitoring and evaluation system application (SIMPATI) has been created and designed to be web-based, facilitating convenient accessibility through a browser. For this application to be connected to the database, the development team employs the PHP (Hypertext Preprocessor) programming language, which is well-suited for server-side scripting and database connectivity. [7][11][12].

Software testing must be conducted during the development process. The objective of software testing is to determine the functionality of the system, identify defects and errors that require correction, and evaluate software features to ensure they meet quality standards [13][14]. System testing employs two testing techniques: manual testing and automated testing. Manual testing is less efficient due to its prolonged duration, susceptibility to human error, and low test accuracy rate [15][16].

Automation testing is a method of reducing the potential for human error in testing, while simultaneously reducing the cost of testing, accelerating the testing process, and improving the accuracy of the testing results [15][17]. To perform automated testing, a testing software program must be utilized to create a test script that simulates the stages of manual test cases in any programming language available in the testing software. This approach eliminates the need for human intervention, thereby reducing the potential for human error [17][18]. One of the tools utilized for automated testing is Katalon Studio[19], which permits the examination of web, mobile, and API-based applications. Additionally, it generates documentation reports of the test results in the form of CSV, XLSX, and PDF [20].

Based on the description mentioned earlier, it is necessary to design a monitoring and evaluation system for higher education by applying the Agile methodology to the SIMPATI application. Additionally, this study evaluates the execution time feature of the application. The evaluation is conducted by implementing automation testing using Katalon Studio.

2. RESEARCH METHODOLOGY

This research method used the agile method in system development and automation testing with the stages shown in Figure 1 below. The literature review is used to understand the context and contribution of previous studies and to build a solid theoretical framework through articles, journals, and other scientific references. The requirements phase is used to define the needs and expectations of the end users for the system to be developed. The requirements phase results in the need to develop the system using the PHP programming language and MySQL database, as well as the need for automated testing using Katalon Studio after the deployment phase.

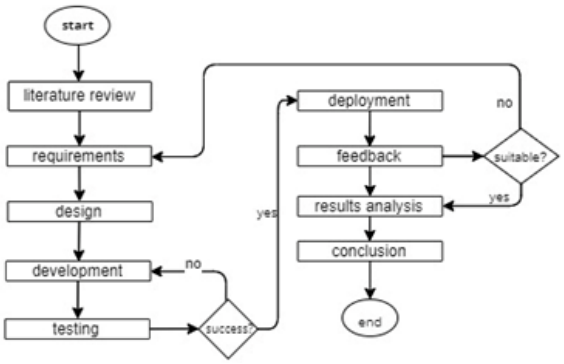

Figure 1. Research flowchart

The description of the stages of the agile method starting from requirements to feedback is presented below.

2.1. Design

The development of the monitoring and evaluation system in this research employs a procedural programming approach. The system design utilizes entity relationship diagrams (ERD) and data flow diagrams (DFD), which are illustrated in the following figure.

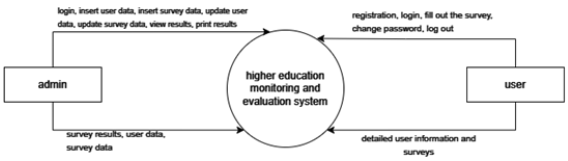

Figure 2. Data Flow Diagram Level 0

Figure 2 illustrates the process flow of the SIMPATI system, which comprises two entities: the administrator and the user. Admin have access to all existing menus, with the exception of the ability to fill out surveys. In contrast, users have the capacity to register accounts, log in, change passwords, fill out surveys, and log out.

Data flow diagram level 1 is shown in Figure 3. The system comprises two roles: the administrator and the user. The user category encompasses lecturers, students, employees, cooperation partners, and alumni. Upon entering an email address and password, the user validates the data. If the data is correct, the user is granted login access.

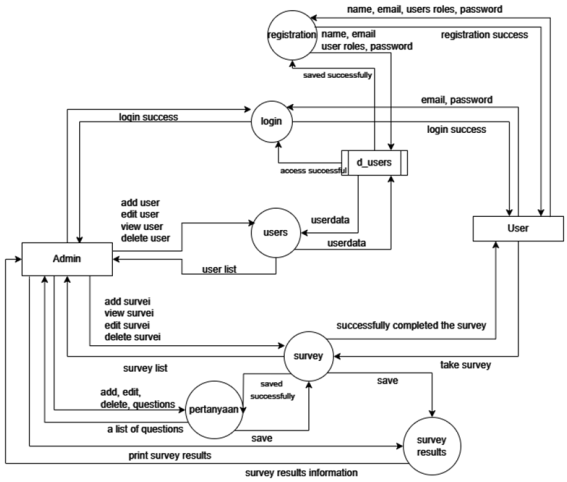

Figure 3. Data Flow Diagram Level 1

Registration is the process of creating a new account on the system, which involves entering the user's name, email address, role within the system, and password. This information is stored in the database. The system displays a list of all registered users, which administrators can access to add, delete, edit, and view user data. The survey process can only be created by administrators, who can also create new surveys, view survey details, edit, and delete surveys. The system will display a list of all surveys that have been created for the administrator. Only users with the appropriate role can complete surveys; the responses will be stored and then displayed in the form of graphs and presentations to administrators who have the right to access survey results and print them.

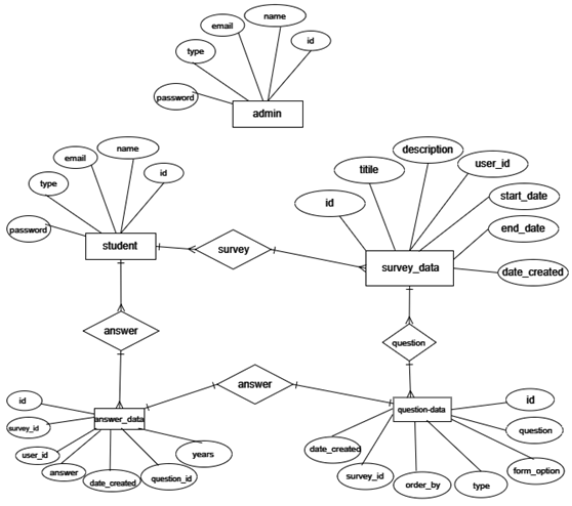

Figure 4. Entity Relationship Diagram

Figure 4 above shows the relationship between entities to be of two types: one-to-one and one-to-many. The one-to-one relationship in this condition is seen from one question that only has one answer. In contrast, the one-to-many relationship is evident in the context of a single user with multiple surveys and responses, as well as in the case of a single survey comprising multiple questions.

In addition to the use of DFD and ERD for database design, the development of a web-based SIMPATI system also utilizes the Figma software for interface design. The following figure presents a view of the SIMPATI dashboard interface design.

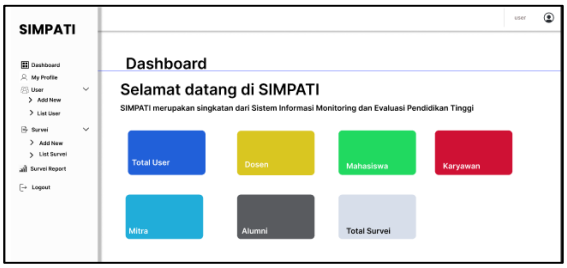

Figure 5. Dashboard interface design

2.2. Development

In this phase of the development, the Agile Development Method is utilized. The system design is converted into lines of program code using Visual Studio Code software. The programming language is PHP version 8, and the database is MySQL with phpMyAdmin software. This phase of development makes use of local server access with Xampp software.

2.3. Testing

Testing is conducted manually and automatically. Manual testing is performed using black box testing. This is conducted before deployment to determine the functionality of the system. Automation testing is carried out after deployment using an automation testing tool, which is repeated automatically. This allows for the observation of the stability of a web-based system that has been deployed.

2.4. Deployment

In this research, the deployment utilizes cPanel at the URL https://simpati.site as a server provider service, with specifications as outlined in the following table.

Table 1. Specification of server

No.	Device	Specification
1	CPU	Intel Core i3 - 5020U @ 2.2GHz
2	RAM	4 GB
3	Storage	120 GB
4	IP Address	202.152.157.51
5	OS	Windows 10 Pro

2.5. Feedback

The feedback stage is employed to obtain feedback. Feedback is generated when the developed application does not align with expectations or contains errors, prompting the developer to implement continuous improvements until the desired outcomes are achieved. Additionally, at the feedback stage, automation testing is conducted to

assess the functionality of SIMPATI deployed. The automation testing process is executed every feature via the Katalon Studio software, with 14 test cases displayed in the following table.

Table 2. List of test-case

No.	Test Case	Features
1	TC-1	Log in success
2	TC-2	Log in failed
3	TC-3	Registration
4	TC-4	New survey
5	TC-5	Edit survey
6	TC-6	View survey
7	TC-7	Delete survey
8	TC-8	New user
9	TC-9	Edit user
10	TC-10	View user
11	TC-11	Delete user
12	TC-12	Survey result
13	TC-13	Print survey
14	TC-14	Forgot password

3. RESULT

3.1. The Result of System Development

The SIMPATI web-based application is utilized by a diverse range of users, including administrators, students, lecturers, employees, cooperation partners, and alumni. Each of these user roles is assigned distinct access privileges. Administrative users are granted the ability to oversee all data within the SIMPATI application, while other user roles are permitted to complete surveys utilized for monitoring and evaluating higher education.

The survey responses submitted by users are stored in the MySQL database in real time, and only administrators are able to view the survey results. The survey results are presented in the form of a percentage graph, which facilitates the evaluation process. The following figure displays the results of the design of the web-based SIMPATI application.

The admin dashboard (Figure 6) can be accessed by administrators who have logged in. This page contains information, including data on the total number of users, lecturers, students, employees, stakeholders, alumni, and the number of surveys conducted. Additionally, a sidebar with menus for the dashboard, my profile, user, survey, survey report, and logout is available.

Figure 6. Admin dashboard

The survey report page (Figure 7) contains information about the survey results that have been input by the user, including the percentage of survey answers and survey descriptions.

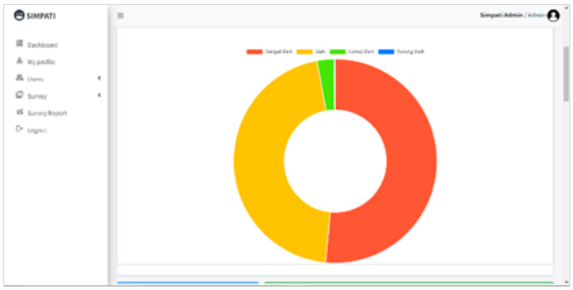

Figure 7. Survey report page

The survey list page for the admin role (Figure 8) contains information on the list of surveys that have been created. Administrators can view survey details through the "View" action, edit surveys if necessary through the "Edit" action, and delete surveys through the "Delete" action. On this page, the administrator can add a new survey through the "Add New" button, which will redirect to the "Add New" page.

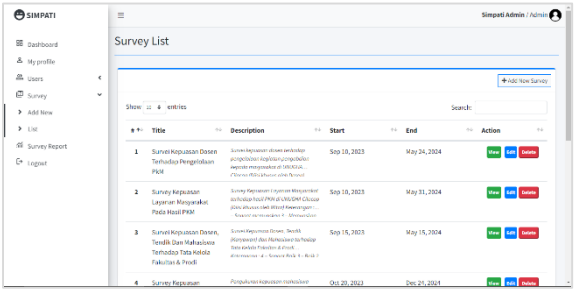

Figure 8. Survey list page

The user list page in the admin user role (Figure 9) contains information about users who have been created in the application. There are view actions to view user details, edit users, and delete users. On this page, the admin can add a new survey by pressing the "Add New User" button, which will redirect to the "Add New User" page.

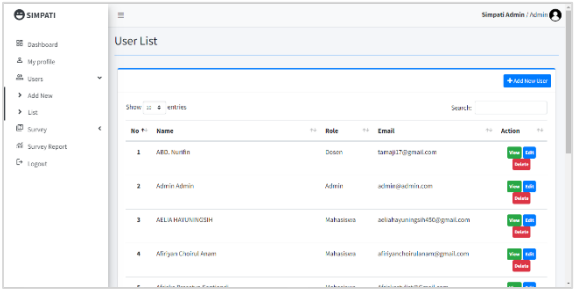

Figure 9. User list page

The user dashboard page (Figure 10) contains information regarding survey completion guidelines, the number of surveys that have been completed, and survey widgets for users. The page includes a sidebar menu, a user profile, a survey list, and a logout option.

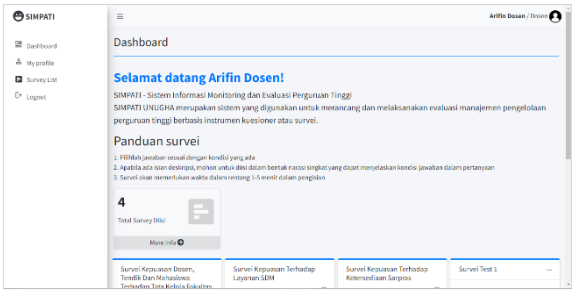

Figure 10. User dashboard page

The page for survey completion by users (Figure 11) contains information regarding the survey description, the year in which the survey was completed, and survey questions that must be completed by the user.

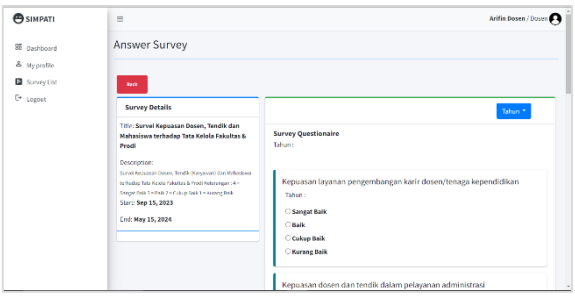

Figure 11. Survey completion page

3.2. Test Result

The automatic testing of SIMPATI web-based applications is conducted using Katalon Studio. This test is employed to ascertain the performance of features within the SIMPATI application. The performance of a feature is indicated by the execution time parameter associated with that feature [21]. Each feature is tested five times, with four test cases passing and one test case failing (shown in Figure 12). The presence of a failed test case is intended to ascertain whether the feature in question is genuinely incapable of functioning if the data component is transmitted incompletely [22].

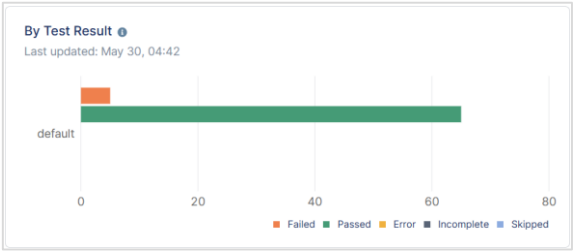

Figure 12. Test Result Summary

The results of automatic testing of each feature in the SIMPATI application are seen through the parameters of the execution result status and execution time. Tests are run at intervals of 4 hours once for 2 days, the test results can be seen in the following table.

Table 3 Tests Result

No.	Test Case	Features	Result	Execution Time
1	TC-1	Log in success	Pass	22s
			Pass	24s
			Pass	29s
			Pass	21s
2	TC-2	Log in failed	Pass	18s
			Pass	17s
			Pass	11s
			Pass	15s
3	TC-3	Registration	Pass	24s
			Pass	46s
			Pass	27s
			Pass	27s
4	TC-4	New survey	Pass	1m.39s
			Pass	1m.42s
			Pass	45s
			Pass	57s
5	TC-5	Edit survey	Pass	31s
			Pass	43s
			Pass	1m.0s
6	TC-6	View survey	Pass	45s
			Pass	28s
			Pass	39s
			Pass	52s
7	TC-7	Delete survey	Pass	46s
			Pass	54s
			Pass	37s
			Pass	1m.54s
8	TC-8	New user	Pass	1m.11s
			Pass	1m.13s
			Pass	51s
			Pass	49s
9	TC-9	Edit user	Pass	4m.29s
			Pass	31s
			Pass	31s
			Pass	39s
10	TC-10	View user	Pass	3m.9s
			Pass	33s
			Pass	24s
			Pass	35s
11	TC-11	Delete user	Pass	2m.10s
			Pass	37s
			Pass	40s
			Pass	43s
12	TC-12	Survey result	Pass	3m.13s
			Pass	26s
			Pass	1m 8s
			Pass	54s
13	TC-13	Print survey	Pass	1m.9s
			Pass	33s
			Pass	35s
			Pass	2m.4s
14	TC-14	Forgot password	Pass	39s
			Pass	25s
			Pass	38s
			Pass	39s
			Pass	22s

The results of the test execution on the table 3, which passed for all test cases, indicate that the system is running stably. However, when the test is run, the response time for each test case varies when the test case is run several times during the interval. The response time indicates the complexity of the test cases. Some test cases run faster, and in the TC-2 test case, the faster response time indicates that TC-2 involves simpler functionality. In contrast, the TC-8 test case exhibits a notable disparity in response time,

reflecting the involvement of more intricate operations and external dependencies. The observed variation in response time can be attributed to a multitude of factors, including system load, network conditions, the complexity of the feature itself, and server conditions.

4. DISCUSSION

The design of the web-based SIMPATI application using the Agile methodology produces a system that can monitor and evaluate higher education. In line with the concept of Agile methodology, system development is carried out quickly with rapid progress and improvement [23]. This application has the main features of multi-user, online surveys, real-time databases, and reports on survey results in the form of graphic visualizations.

Figure 12 shows that the test results were as expected. There were 65 test cases with successful execution results and 5 test cases with unsuccessful results. The average execution time of each feature was calculated based on the results in Table 3 and Figure 13. The average execution time of less than one minute (60 seconds) is observed in simple features, such as Login, Register, View, and Forgot Password. In contrast, features such as Add New Survey, Add New User, Edit User, Edit Survey, Delete User, Delete Survey, and Print Survey generate an average execution time exceeding 60 seconds.

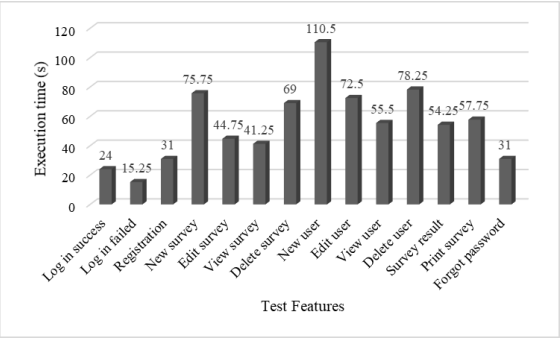

Figure 13. Execution time every feature in SIMPATI web-app

Based on the system design in Figure 3 (DFD Level 1), these features exhibit a more complex data flow. Consequently, the execution time is also longer than that of features with a simple data flow. The fastest execution time is observed in the Login feature, while the longest execution time is observed in the feature of adding a new user. This is because the feature of adding a new user is performed by the admin role. The new user data is transmitted to the database with an access relation to the survey feature, and then the new user data is displayed back to the interface.

5. CONCLUSION

The design and implementation of the web-based SIMPATI application produces a system that can monitor and evaluate higher education. The Agile development method was employed to create this application, which features the following main features: multi-user, online survey, real-time database, and data visualization. Automated testing using Katalon Studio demonstrated that all features in the system run well and are stable. Nevertheless, there is a discrepancy in the execution time for each test case. This is attributed to the intricacy of the test cases and external factors such as system load, server, and network conditions. The fastest execution time for the login feature is 11 seconds, while the longest execution time for the add new user feature is 4 minutes and 29 seconds.

REFERENCES

[1] F. P. Utami, F. A. Artanto, and M. Y. Febrianto, "Sistem Survey Pelayanan Masyarakat Di Kantor Sekretariat Daerah Kabupaten Pekalongan Berbasis Web," *Surya Inform.*, vol. 13, no. 1, pp. 1–9, 2023.

[2] D. L. Kaligis and R. R. Fatri, "Pengembangan Tampilan Antarmuka Aplikasi Survei Berbasis Web Dengan Metode User Centered Design. JUST IT : Jurnal Sistem Informasi, Teknologi Informasi Dan Komputer, 10(2), 106. https://doi.org/10.24853/justit.10," *JUST IT J. Sist. Informasi, Teknol. Inf. dan Komput.*, vol. 10, no. 2, p. 106, 2020.

[3] A. F. Rahmawati, H. Tolle, and R. I. Rokhmawati, "Pengembangan Sistem Informasi Monitoring Dan Evaluasi Hasil Kegiatan Pengawas Berbasis Web (Studi Kasus : Dinas Pendidikan Kota Malang)," vol. 3, no. 3, pp. 2452–2458, 2019.

[4] D. A. Megawaty, "Sistem Monitoring Kegiatan Akademik Siswa Menggunakan Website," *J. Tekno Kompak*, vol. 14, no. 2, p. 98, 2020, doi: 10.33365/jtk.v14i2.756.

[5] C. Aditya Wijaya, "Sistem Monitoring dan Evaluasi Pengelolaan Program Studi di Institusi Pendidikan Tinggi," *Indones. J. Inf. Syst.*, vol. 1, no. 1, pp. 13–24, 2018, doi: 10.24002/ijis.v1i1.1723.

[6] O. M. Febriani, A. S. Putra, and R. P. Prayogie, "Rancang Bangun Sistem Monitoring Sirkulasi Obat Pada Pedagang Besar Farmasi (PBF) Di Kota Bandar Lampung Berbasis Web," pp. 122–132, 2020.

[7] M. J. Samad, Tommy Dwi Cahyono, "SISTEM MONITORING DAN EVALUASI KEGIATAN BELAJAR MENGAJAR(E-MONEV KBM) BERBASIS WEB," vol. 1, no. 2, pp. 169–176, 2019.

[8] F. Nurzaman, "Pengembangan Sistem Otomatisasi Tagihan Menggunakan Metode Agile Software Development," *J. IKRA-ITH*

Inform., vol. 4, no. 1, pp. 46–57, 2020.

[9] T. Ayunita Pertiwi *et al.*, "Perancangan Dan Implementasi Sistem Informasi Absensi Berbasis Web Menggunakan Metode Agile Software Development Web-Based Attention Information System Design and Implementation Using the Agile Software Development Method," *J. Test. dan Implementasi Sist. Inf.*, vol. 1, no. 1, pp. 53–66, 2023.

[10] R. Gutama and T. Dirgahayu, "Implementasi Scrum Pada Manajemen Proyek Pengembangan Aplikasi Sistem Monitoring dan Evaluasi Pembangunan (SMEP)," *J. Autom.*, vol. 3, no. 1, pp. 29–35, 2021.

[11] A. Mubarak, "Rancang Bangun Aplikasi Web Sekolah Menggunakan Uml (Unified Modeling Language) Dan Bahasa Pemrograman Php (Php Hypertext Preprocessor) Berorientasi Objek," *JIKO (Jurnal Inform. dan Komputer)*, vol. 2, no. 1, pp. 19–25, 2019, doi: 10.33387/jiko.v2i1.1052.

[12] A. Herdiansah, R. I. Borman, and S. Maylinda, "Sistem Informasi Monitoring dan Reporting Quality Control Proses Laminating Berbasis Web Framework Laravel," vol. 15, no. 2, pp. 13–24.

[13] D. Debiyanti, S. Sutrisna, B. Budrio, A. K. Kamal, and Y. Yulianti, "Pengujian Black Box pada Perangkat Lunak Sistem Penilaian Mahasiswa Menggunakan Teknik Boundary Value Analysis," *J. Inform. Univ. Pamulang*, vol. 5, no. 2, p. 162, 2020, doi: 10.32493/informatika.v5i2.5446.

[14] K. Anita, A. D. Wahyudi, and E. R. Susanto, "Aplikasi Lowongan Pekerjaan Berbasis Web Pada Smk Cahaya Kartika," *J. Teknol. dan Sist. Inf.*, vol. 1, no. 1, pp. 75–80, 2020, doi: 10.33365/jtsi.v1i1.213.

[15] D. Ateşoğulları and A. Mishra, "Automation Testing Tools: a ComparativeView," *Int. J. Inf. Technol. Secur. №*, vol. 4, no. December, p. 2020, 2020.

[16] A. Samad, T. Nafis, S. Rahmani, and S. S. Sohail, "A Cognitive Approach in Software Automation Testing," *SSRN Electron. J.*, 2021, doi: 10.2139/ssrn.3834262.

[17] Dewandra Sapto Prasetyo and W. Silfianti, "Analisis Perbandingan Pengujian Manual Dan Automation Testing Pada Website E-Commerce," *J. Ilm. Tek.*, vol. 2, no. 2, pp. 127–131, 2023, doi: 10.56127/juit.v2i2.516.

[18] J. L. Min, A. Istiqomah, and A. Rahmani, "Evaluasi Penggunaan Manual Dan Automated Software Testing Pada Pelaksanaan End-To-End Testing," *JTT (Jurnal Teknol. Ter.*, vol. 6, no. 1, p. 18, 2020, doi: 10.31884/jtt.v6i1.256.

[19] E. Pelivani and B. Cico, "A comparative study of automation testing tools for web applications," *2021 10th Mediterr. Conf. Embed. Comput. MECO 2021*, pp. 7–10, 2021, doi: 10.1109/MECO52532.2021.9460242.

[20] V. T. Tempomona, U. Kristen, and S. Wacana, "Penerapan Metode Blackbox Pada Perangkat Lunak Menggunakan Katalon Studio (Studi Kasus : Aplikasi Absensi di PT Astra Sedaya Finance)," pp. 193–204, 2022.

[21] V. Safaat, G., & Utami Tjhin, "Analysis of Quality Assurance Performance in the Application of Manual Testing and Automation Testing for Software Product Testing," *Indones. Interdiscip. J. Sharia Econ.*, vol. 7, no. 2, pp. 1987–1996, 2024, [Online]. Available: https://e-journal.uac.ac.id/index.php/iijse/article/view/4561/1693.

[22] R. Faizal, Berliyanto, A. Nurrohman, and R. Setiabudi, "COMPARATIVE ANALYSIS OF AUTOMATION FUNCTIONAL TESTING," *Techno Nusa Mandiri J. Comput. Inf. Technol.*, vol. 21, no. 1, pp. 9–14, 2024.

[23] S. Al-Saqqa, S. Sawalha, and H. Abdelnabi, "Agile software development: Methodologies and trends," *Int. J. Interact. Mob. Technol.*, vol. 14, no. 11, pp. 246–270, 2020, doi: 10.3991/ijim.v14i11.13269.