Published *online* on the journal's web page : http://innovatics.unsil.ac.id

Innovation in Research of Informatics (INNOVATICS)

| ISSN (Online) 2656-8993 |

Software Testing in the Indonesian Industry: Survey of Methods, Tools, and Documentation

Asri Maspupah[1], Ani Rahmani[2], Joe Lian Min[3], Trisna Ari Roshinta[4]

[1,2,3]*Department of Computer and Informatics Engineering, Politeknik Negeri Bandung, Jl. Gegerkalong Hilir, Bandung 40559, Indonesia*
[4]*Computer Engineering, Budapest University of Technology and Economics, Budapest, Műegyetem rkp. 3, 1111, Hungaria*

[1]*asri.maspupah@polban.ac.id,* [2]*anirahma@polban.ac.id,* [3] *joelianmin@polban.ac.id,* [4] *trisna.roshinta@edu.bme.hu*

ARTICLE INFORMATION

Article History:
Received: September 9, 2024
Last Revision: September 27, 2024
Published *Online*: September 30, 2024

KEYWORDS

Software Testing,
Survey an Industry Software,
Software Testing Methodology,
Software Testing Tools,
Software Testing Documentation

CORRESPONDENCE

Phone: +6285222496028
E-mail: asri.maspupah@polban.ac.id

ABSTRACT

Software testing plays a crucial role in the software development by ensuring that software is accurate and of high quality. Many software companies neglect software testing, which can lead to unprofitable business outcomes. For example, ineffective software testing may fail to identify all defects, resulting in increased development costs. A key factor determining the success of software testing is the strategy for implementing the testing process, the selection of testing tools, and the documentation of testing activities. This article examines the current state of software testing processes in the Indonesian software industry. The research objective is to analyze the software testing implementation strategy within the software development context, focusing on three main areas: software testing methodology, software testing tools, and software testing documentation. The research employs a survey method, collecting data from several respondents, Indonesian software companies, via an online questionnaire. The research findings indicate that testing is still predominantly manual. However, some software companies have begun to adopt a combination of manual and automated testing. Most software companies conduct testing at all levels and prepare testing scenarios as guidelines for executing test cases. Additionally, they utilize software testing documentation for reporting purposes during the execution of tests. Nevertheless, documenting test cases as a guide for testing execution is not prioritized as highly as bug reporting. Conversely, many Indonesian software companies have adopted testing tools and conducted performance testing to ensure software quality. Consequently, the software testing process in the Indonesian software industry tends to adhere to formal methods in accordance with the ISO/IEC/IEEE 29119 software testing standards.

1. INTRODUCTION

Software testing is a systematic approach to evaluating software quality in relation to product standards, procedural standards, and software development process standards [1]. It serves as a key indicator for identifying software defects, clarifying ambiguities in requirements, and improving product quality through verification and validation processes. Additionally, software testing enhances both reliability and usability [2]. Therefore, software testing plays a crucial role in the software development process by identifying issues and challenges and ensuring the delivery of a high-quality product [3].

The implementation of software testing consists of three primary activities: test planning, testing strategies, and the utilization of testing tools [4]. According to WG26, the standard software testing process encompasses a variety of techniques and approaches. Each technique possesses its own strengths and weaknesses, making the selection of a software testing technique highly dependent on the characteristics of the software being evaluated. Optimizing the software testing process can enhance the quality of various types of software [5]. Different software development models, types of software, and levels of software complexity necessitate the use of distinct software

testing techniques. However, software testing stages currently do not receive as much attention as other phases of software development [1], [6], and are often overlooked in the software industry [7]. This oversight is primarily due to customers perceiving software testing as an unnecessary cost, which results in the software industry lacking sufficient time to conduct thorough testing [8]. Meanwhile, the execution of software testing requires 50% of the total available time [9]. The implementation of software testing remains immature and is typically conducted on an ad hoc basis. This leads to various unfavorable outcomes, such as the failure to detect all defects or ineffective testing processes, with costs escalating due to exceeding the allocated testing time [10].

Several issues hinder the development of the software testing process in the software industry across various countries. In Garousi's research, which surveyed software testing implementation in Austria, Canada, Denmark, Germany, Hungary, Israel, Sweden, and Turkey, the identified challenges included a lack of test documentation, numerous test cases with identical testing objectives, a predominant reliance on manual testing methods—particularly for repetitive tasks—insufficient test management documentation, ineffective metrics for prioritizing testing efforts, and difficulties in utilizing testing tools [10]. On the other hand, issues related to software tester compliance in countries such as China, Cuba, and India highlight the lack of knowledge among software testers regarding standard software testing processes [11]. These compliance challenges encompass the analysis of testing objectives based on requirements, varying testing environments among software testers, a lack of creativity, and the execution of tests that do not align with the test design.

Meanwhile, the implementation of software testing in Indonesia has received limited attention from researchers. One of the few studies conducted by Hariyanto concluded that small and medium-sized software companies in the Yogyakarta and Tangerang Selatan regions have adopted software testing processes, focusing on the number of testers involved and the utilization of testing tools [12]. On the other hand, Wahyuningrum investigated the software testing process in Indonesia by reviewing various papers published by Indonesian researchers. The scope of the reviewed literature included observations from researchers who adopted the standardized ISO/IEC/IEEE 29119 software testing processes. The results of her study indicate that, as of the end of 2016, researchers in Indonesia have begun to explore the implementation of the ISO/IEC/IEEE 29119 standard in software testing practices [13].

Based on the issues, researchers have not thoroughly examined the implementation of the ISO/IEC/IEEE 29119 standardized software testing in Indonesia's software industry. However, testing is a crucial component in ensuring software quality, and failing to detect bugs early can lead to significant costs during maintenance. Several factors that may hinder the success of software testing include a lack of management support, unclear testing methodologies, and limited resources [14]. Therefore, this research aims to investigate how the software industry in Indonesia practices software testing processes.

The research aims to identify challenges related to software testing methodologies, tools, and documentation in Indonesia. Consequently, the Research Questions (RQs) for this study are: (1) to determine the key variables that influence the execution of software testing concerning methodologies, tools, and documentation; and (2) to assess the current state of software testing in the Indonesian software industry. The proposed research solution focuses on enhancing testing practices within Indonesia's software industry by aligning them with standardized software testing processes. This initiative encompasses various software testing methodologies, tools, and documentation. The findings of this research will improve the practical skills of software testers in Indonesia by providing insights into effective methods, tools, and documentation for conducting tests. Additionally, it will deepen the academic understanding of standardized software testing processes.

The research methodology follows Panto's survey research approach, which includes developing survey instruments, selecting sample respondents, collecting data, and analyzing the results. The survey instruments are designed based on the research scope, which includes software testing methodologies, tools, and documentation. Sample respondents are selected from software companies located in major Indonesian cities, such as Jakarta and Surabaya, to ensure a representative sample of the industry. Data collection is conducted through online questionnaires to expand the reach of the data gathering process. Data analysis evaluates the distribution of software testing practices based on the variables within the software testing scope, following ISO/IEC/IEEE 29119, as outlined in the study.

This scientific article is organized into five sections. The first section discusses the background of the study. The second section reviews previous studies relevant to the objectives of this research. The third section outlines the research methodology employed. The fourth section presents the results of a survey on the state of software testing in the Indonesian software industry, followed by a discussion of the research findings. Finally, the conclusions are presented in the fifth section.

2. RELATED WORK

Research on software testing surveys in practical industries has been conducted by Hynninen, Wang and Santos. Hynninen explored the current state of software testing by involving 33 companies in Finland.. The study examined how software organizations evaluate their products and the process models they implement. Hynninen then compared the results of the 2017 survey with those from 2009. This comparison revealed a significant shift in organizations towards test automation, moving away from manual testing and adopting more advanced testing infrastructures. Additionally, most software industries have adopted agile practices and reduced their reliance on formal process models [15].

Meanwhile, Wang conducted a survey involving 151 practitioners from 101 organizations across 25 countries in Europe to assess the maturity level of automation testing implementation within the software testing process. The survey results indicated significant variability in maturity levels, with 85% of practitioners reporting that their testing teams had sufficient skills. However, 47% acknowledged a lack of guidelines for designing and executing automated

tests. Several factors in the software development process can influence the maturity level of automation testing implementation, including the adoption of Agile and/or DevOps development models, a higher percentage of automated test cases, and the complexity of the software. In his research, Wang concluded that while test automation is advancing, many organizations have yet to achieve full maturity in their automation testing processes [14].

On the other hand, Santos conducted a survey of 136 practitioners in the Brazilian software industry regarding the implementation of the software testing process. The survey results revealed that system testing is the most utilized type of testing. Additionally, many software companies perform critical regression tests after software updates to verify whether new changes impact the unchanged components of the software. In the realm of automated testing, Selenium is the most frequently employed tool. The Brazilian software industry also faces three main challenges: (1) a lack of prioritization in testing; (2) limited knowledge and training in testing; and (3) difficulties in the automation testing process. In his research, Santos concluded that most companies in Brazil continue to encounter challenges in implementing software testing, particularly concerning the adoption of automation techniques and tools. This study emphasizes that collaboration between industry and academia is essential for enhancing testing methods, as well as the need for increased investment in research and training for software testers [7].

Lastly, Hariyanto conducted a survey on the implementation of software testing within the software industry in the Yogyakarta and South Tangerang regions of Indonesia. The survey instruments utilized by Hariyanto included questions regarding the number of software testers involved in the testing process, the application of software testing methodologies, the use of manual testing, and the employment of testing tools. The respondents in this study were drawn from small and medium-sized software industries. The research findings indicated that these small and medium-sized software industries have adopted software testing practices with varying degrees of implementation, including manual testing approaches, the use of testing tools, and the number of software testers involved in the testing process [12].

Based on four related studies, the survey regarding the implementation of the software testing process in Indonesia has not been thoroughly explored in existing research. The fourth study examined the implementation of software testing in Indonesia; however, it did not utilize a wide range of survey instrument variables. The variables included only the number of software testers, whether software testing was conducted, and whether testing tools were employed. This article surveys the software industry in Indonesia to examine its software testing practices aimed at ensuring software quality. It utilizes a survey instrument that focuses on software testing methodologies, testing tools, and testing documentation. Furthermore, the survey results are compared with the software testing process standards outlined in ISO/IEC/IEEE 29119, to draw conclusions about whether the software testing process complies with these standards or if it still falls short. The scope of this survey is limited to software quality assurance at the end of the product implementation phase, with a focus on the software testing life cycle. As a result, there may be gaps in the survey findings, such as the implementation of software testing during the requirements analysis phase, software design, the bug life cycle employed, and communication throughout the software testing process.

One method for conducting this survey is survey research, which involves data collection techniques that include distributing survey instruments to a sample of respondents. The developed survey instrument is validated by assessing measurement error. This validation process includes domain experts completing the survey and evaluating its accuracy concerning the topic under investigation. Enhancements to the survey instrument are based on its effectiveness in capturing information within a specific domain. A challenge for researchers utilizing survey research is selecting a sampling approach that ensures the sample accurately represents the population. Furthermore, a potential drawback of using survey research is that the results may lack relevance if the respondents are not domain experts.

3. METHODOLOGY

The research method employs a survey research approach as introduced by Ponto. This method was selected because it aligns with the characteristics of the research, which aims to observe the state of software testing implementation in the Indonesian software industry. The study relies on expert judgment from industry practitioners who can accurately represent the field. Survey research is defined as the collection of information from a sample of respondents through their responses to questions formulated in survey instruments [16]. The data collection strategy within the survey research method can combine both quantitative and qualitative approaches, providing flexibility based on the research objectives. The data collection strategy employed in this study combines both quantitative and qualitative approaches. The quantitative component utilizes structured survey instruments that gather numerical data through closed-ended questions. On the other hand, the qualitative approach incorporates open-ended questions, allowing for a deeper exploration of diverse perspectives. This combination of data collection strategies is characterized by questions that are neither entirely open-ended nor completely closed-ended. Nevertheless, most of the statements in the survey instruments predominantly reflect the quantitative strategy.

3.1 Development of Survey Instruments

The software testing survey instrument is designed based on the software testing process objectives identified during the literature review. The survey instrument is structured as a matrix that maps testing methodologies, tools, and documentation to each testing strategy within software development. Each fundamental objective of software testing is translated into questions aimed at collecting data on software testing trends in the Indonesian software industry. The evaluation of the software testing survey instrument involves the following steps: (1) ensuring that the questions are clear and comprehensible, (2) assessing the response rate to maximize the likelihood of effective follow-up procedures, (3) evaluating the

validity of the survey instrument by verifying the alignment of the questions' objectives with established software testing process standards, and (4) ensuring the appropriateness of the data analysis techniques and expected responses [7]. Additionally, the survey instrument includes questions about the profiles of software industry organizations to identify the respondents. The stages of developing a survey instrument are outlined as follows:

1) Analysis of the Results from the Literature Review

The implementation of software testing is closely linked to the methods employed at each stage of testing. Generally, these implementation methods are categorized based on specific domains, as illustrated in Table 1.

TABLE 1. IMPLEMENTATION METHODS OF SOFTWARE TESTING [17]

Domain	Implementation Methods
Testing methodologies	Black Box Testing (Behavior Testing), White Box Testing (Structural Testing), Exploratory Testing, Test Level (Unit Test, Integration Test, System Test, User Acceptance Test), Manual Testing, Automation Testing, Dynamic Testing, Static Testing, Test Type (Functional Testing, Regression Testing, Graphical User Interface Testing, Load Testing, Database Testing, Compatibility Testing)
Testing techniques	Equivalence Class Partitioning, Boundary value analysis, Data Cycle Test, Error Guessing, Comparison Testing, Cause-Effect Diagram, Data Flow, Path, Statement, Loop Coverage, Mutation Testing, Logic & Fault Based, Domain & Boundary
Testing tools	Test Automation, Performance Testing, Test Management

Table 1 illustrates the grouping and implementation process of software testing into three categories: testing methodologies, testing techniques, and testing tools. Each category may encompass various strategies for the software testing process. In his research, Izzat classified the implementation of the software testing process according to the standards established by ISO/IEC/IEEE 29119.

On the other hand, the implementation of software testing is significantly influenced by the testing strategies employed by software testers, as well as the skills required in the software industry. A testing strategy refers to a systematic approach to testing methods that facilitates effective testing, ranging from component-level and low-level tests to integration-level and high-level tests [18]. Currently, there are international standards governing software testing processes. The ISO/IEC/IEEE 29119 standard provides a comprehensive overview of the software testing process within the context of the software development life cycle [4], as illustrated in Figure 1.

Figure 1 illustrates the application of generic sub-processes across different test levels/phases and test types. The diagram in Figure 1 also explains the relationship between test types and quality characteristics, with each test type targeting a specific quality characteristic. The technical skills of a software tester in the field of software testing can be categorized into three areas: testing skills, information technology (IT) skills, and domain knowledge. Testing skills refer to the software tester's ability to perform activities related to the software testing life cycle, including test case design, test execution, and test reporting. Meanwhile, IT skills involve the software tester's ability to utilize information technology during the software testing process, such as employing testing tools or preparing the testing environment. Lastly, domain knowledge refers to the software tester's understanding of the requirement specifications established in the early phases of software development, which defines the scope of testing [19].

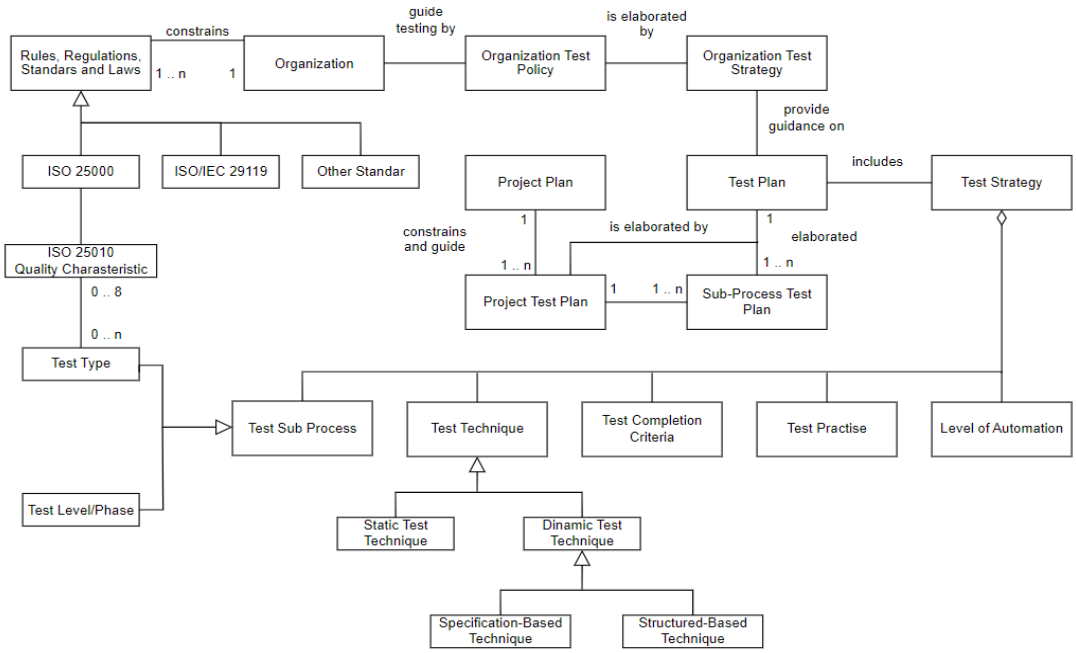

FIGURE 1. TESTING PROCESS IN THE CONTEXT OF A PROJECT [4]

In 2014, Reid developed a model for the technical skills required in software testing using the Personal Test Maturity Matrix (PTMM) framework. The PTMM framework is structured around three key components: testing jobs, test roles, and test skills [19], as illustrated in Figure 3. A testing job encompasses the specific software testing tasks associated with each software product. A test role involves various strategic approaches to conducting

software testing within a given testing job. Test skills refer to the specific abilities needed to effectively perform a test role within a testing job.

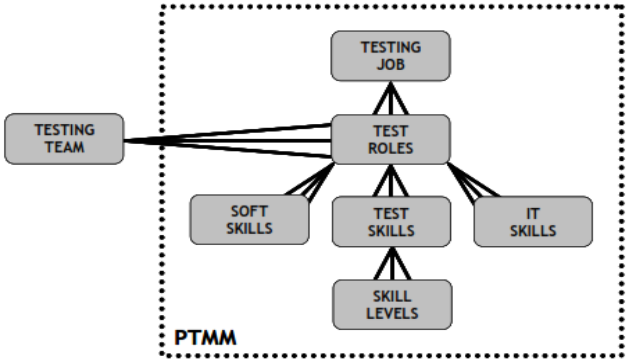

FIGURE 2. THE STRUCTURE OF THE PTMM FRAMEWORK [19]

Figure 2 illustrates that each testing team comprises multiple test roles, each responsible for various testing jobs. Each test role will have more than one soft skill, testing skills, and IT skills. Additionally, the proficiency in each testing skill has different skill levels depending on the individual's experience in executing software testing.

Based on the findings from the literature review, the variables influencing software testing practices—specifically methodologies, tools, and documentation, as illustrated in Table 1, Figure 1, and Figure 2—can be identified using the PTMM. Table 1 outlines the domains of software testing implementation, accompanied by examples of methodologies, tools, and documentation. Figure 1 provides a comprehensive overview of the software testing methodology domain, categorizing software testing approaches into test levels, test design techniques, test execution, and performance testing. Meanwhile, Figure 2 illustrates the mapping of influential variables using the PTMM. Each piece of literature on software testing practices, as referenced in Table 1 and Figure 2, is aligned with testing jobs, roles, and skills to formulate survey instrument questions, as detailed in Table 2. Consequently, the variables influencing software testing practices include test levels, test design techniques, test execution, performance testing, testing tools, and test documentation. The testing tools employed encompass those that assist in performance testing, test automation, and bug reporting. Additionally, test documentation refers to the materials utilized during test case development, test case execution, and bug reporting.

2) Preparation of the Question List

The purpose of creating the questionnaire is to quantitatively assess the software testing process based on the various testing roles performed in the software industry. Based on the ISO/IEC/IEEE 29119 software testing standards, six categories of test roles have been identified within the software testing life cycle: testing tools, test levels, test design techniques, test execution, performance testing, and test documentation. Each test role encompasses a specific set of test skills, IT skills, and domain knowledge. In this study, these skills and knowledge areas pertain to the technical testing capabilities within the scope of software testing methodologies, tools, and documentation. The mapping of test roles to technical testing capabilities is presented in Table 2.

TABLE 2. MAPPING OF TECHNICAL TESTING CAPABILITIES BASED ON PTMM FRAMEWORK

Testing Job	Testing Role	Implementation Methods
Methodology	Test level	Unit testing, integration testing, system testing dan accepted testing
Methodology	Test design technique	Structured base/ white box, specification based/ black box, experience based
Methodology	Test execution	Static testing, dynamic testing, manual testing dan automation testing
Methodology	Performance testing	Load testing, volume testing, stress testing, volume testing, compatibility testing, interoperability testing dan portability testing
Tools	Testing tools	Load testing tools, test automation testing tools, dan bug/defect reporting tools
Documentation	Test documentation	Test design specification and test procedure specification documentation, dan test execution documentation

The technical testing capabilities presented in Table 2 serve as the foundation for the questionnaire designed to collect information regarding the implementation of the software testing process. This is based on the application of testing strategies, assessed through closed-ended questions. In addition to closed-ended questions, this research incorporates open-ended questions to identify other testing tools utilized within the scope of testing tool inquiries. These open-ended questions are qualitative, allowing respondents to list the testing tools used in the software industry without being restricted to the provided options. Meanwhile, the answer choices for testing tools, such as Selenium, Postman, and Jira, in questions four through eight of the survey instruments were selected based on popular testing tools commonly used, with references from the website https://www.softwaretestingmaterial.com/. A list of survey instrument questions related to software testing methodologies, tools, and documentation in Table 3.

3.2 Selection of Respondent Sample

Selecting a representative sample from the population is crucial to ensure that the results can be generalized. Large random samples are typically employed to accurately reflect the characteristics of the population. In this study, participants from the Indonesian software industry serve as research respondents. Ideally, all software companies across Indonesia would be surveyed for this research. However, due to the uneven distribution of the industry and the challenges associated with reaching a nationwide population, a more targeted approach was adopted. Previous research on software testing practices in developing countries indicated that sample sizes ranging from 25 to 47 organizations were sufficient [20]. Therefore, this study selected respondents from a specific population. The selection criteria focused on software companies located in major cities such as Jakarta and Bandung, as these cities are representative of Indonesia's software industry. Respondents were identified by collecting contact information for IT personnel, including senior software quality assurance specialists, project managers, and lead software engineers, who possess knowledge of the software testing processes within their organizations.

3.3 Data Collection Techniques

An online questionnaire serves as the data collection technique. The questions included in the questionnaire are derived from a software testing survey instrument. It employs closed-ended questions with nominal categories, as well as open-ended questions that allow for "other" responses. Nominal categories refer to response options that do not have a relationship among the set of responses to a given questionnaire question [21]. A Google Form will be utilized to distribute the online questionnaire to the selected respondents via email and WhatsApp.

3.4 Data Analysis of Survey Responses

The data analysis of the completed questionnaires involves examining the comprehensiveness of software testing methodologies, tools, and documentation within each testing strategy across various software industries. Quantitative measurements are then conducted by analyzing the percentage distribution of responses from all participants. Closed-ended questions can be analyzed quantitatively, while semi-closed questions—specifically questions four through eight—must first be converted into quantitative data. This transformation involves summarizing the various testing tools mentioned by respondents. Next, data cleaning is performed to identify unique testing tools and correct any typographical errors. Finally, the process includes calculating the testing tools utilized in the software industry using a quantitative approach. Data analysis will be conducted using Microsoft Excel, utilizing formulas to calculate the percentage of software testing process implementation within the software industry.

Next, the process continues with a comprehensive description of the current state of software testing implementation, informed by the results of the quantitative measurements. The survey findings are presented by examining the least and most prevalent testing approaches and comparing their distributions. Ultimately, a conclusion is reached regarding the alignment of software testing process implementation in the Indonesian software industry with the ISO/IEC/IEEE 29119 standard, particularly in the areas of software testing methodology, tools, and documentation.

TABLE 3. SURVEY INSTRUMENT FOR THE SOFTWARE TESTING PROCESS

No.	Testing Role	Question	Type of Answer	Answer Options
1	-	Does your company conduct testing on the software that is currently under development?	Multiple Choice (only one answer is allowed)	a. Yes b. No
2	Test level	Are the following testing approaches implemented in your company?	Multiple Choice (more than one answer is allowed)	a. Unit Testing b. Integration testing c. System Testing d. User Acceptance Testing
3	Testing tools	Are the following testing tools utilized in your company?	Multiple Choice (more than one answer is allowed)	a. Load testing tools b. Test Automation Tools for Web Applications c. Test Automation Tools for Mobile Applications d. Api testing tools e. Bug management report tools
4	Testing tools	Please indicate the load testing tools currently utilized by your company. You may select more than one answer.	Multiple Choice (more than one answer is allowed)	a. JMeter b. LoadRunner c. Locust d. Not Utilizing Testing Tools e. Other: .....
5	Testing tools	Please select all test automation web application tools currently utilized by your company. You may choose more than one answer.	Multiple Choice (more than one answer is allowed)	a. Catalon Studio b. Cypress c. Selenium d. Cucumber e. Not Utilizing Testing Tools f. Other: .....
6	Testing tools	Please select all test automation mobile application tools currently used by your company. You may choose more than one answer.	Multiple Choice (more than one answer is allowed)	a. Catalon Studio b. Apium c. XCUITest (for iOS) d. Espresso (for Android) e. Not Utilizing Testing Tools f. Other: .....
7	Testing tools	Please select from the list of bug API testing tools currently utilized by your company. You may select more than one answer.	Multiple Choice (more than one answer is allowed)	a. Postman b. SoapUI c. Rest-Assured d Not Utilizing Testing Tools e. Other: .....
8	Testing tools	Bug management report tools currently used by your company you may select more than one answer.	Multiple Choice (more than one answer is allowed)	a. Spreadsheet b. Trello c. Jira d. Mantis e. Github f. Test rail g. Not Utilizing Testing Tools h. Other: .....
9	Test design technique	Are the following test case design approaches utilized in your company?	Multiple Choice (more than one answer is allowed)	a. Black Box Testing b. White Box Testing c. Experience Based

No.	Testing Role	Question	Type of Answer	Answer Options
10	Test execution	Are the following test execution approaches implemented in your company?	Multiple Choice (more than one answer is allowed)	a. Static Testing b. Dynamic Testing c. Automation Testing d. Manual Testing
11	Performance Testing	Are the following application performance testing approaches implemented in your company?	Multiple Choice (more than one answer is allowed)	a. Load Testing b. Stress Testing c. Volume Testing d. Combability Testing e. Interoperability Testing f. Portability Testing
12	Test Documentation	Are the following testing documentation practices implemented in your company?	Multiple Choice (more than one answer is allowed)	a. Test scenario b. Test execution c. Bug reporting

4. RESULT AND DISCUSSION

The questionnaire included in the survey instrument focuses on the implementation of the software testing process within the Indonesian software industry. Additionally, the survey instrument gathers general information about the respondents' profiles, including the organization's size, the number of software developers, and the number of software testers involved in software development. The target respondents are in major cities and represent a variety of organizational scales. This section presents the survey results collected from May to June 2024. A total of 57 organizations participated, with company locations distributed across eight regions: (1) Bandung, with 30 industries; (2) Jakarta, with 17 industries; (3) Cimahi, Cirebon, and Tangerang, with 2 industries each; and (4) Depok, Karawang, Payakumbuh, and Bali, with 1 industry each.

The respondent profile information based on organizational scale is categorized by the number of employees in the software industry, specifically into very small, small, medium, large, and large or very large organizations [15]. Very small organizations represent the smallest segment, comprising approximately 5.26% of the total software industry. In contrast, large and very large organizations are the most prevalent, accounting for about 42.11%. The remaining segments include small-scale organizations at 28.07% and medium-scale organizations at 24.56%. The respondent profile of the software industry by organizational scale is presented in Table 4.

TABLE 4. RESPONDENT PROFILE OF THE SOFTWARE INDUSTRY BY ORGANIZATIONAL SCALE (N = 57)

No.	Category	% From Respondent
1	Very small organization (1-10 employees)	5,26%
2	Small organization (11-50 employees)	28,07%
3	Medium organization (51-250 employees)	24,56%
4	Large or very large organization (250+ employees)	42,11%

The respondent profile information regarding the number of software developers and software testers involved in software development within the industry aims to understand the ratio between the individuals implementing code and those testing the software for compliance with requirements. The survey results indicate that most software companies have development teams with over 50 employees (36.84%), while smaller teams of 1-5 members are less common (8.77%). Meanwhile, medium-sized teams are frequently observed in the software industry, with 31.25% consisting of 6-25 members and 22.81% comprising 26-50 members. The respondent profile of software developers in the industry is presented in Table 5.

TABLE 5. PROFILE OF RESPONDENTS IN THE SOFTWARE INDUSTRY, NUMBER OF SOFTWARE DEVELOPERS

No.	Category	% From Respondent
1	The number of software developers ranges from 1-5 people	8,77%
2	The number of software developers ranges from 6-25 people	31,25%
3	The number of software developers ranges from 26-50 people	22,81%
4	The number of software developers is 50+ people	36,84%

On the other hand, Table 6 presents the survey results regarding the profiles of respondents and the number of software testers involved in software development within the software industry. The findings indicate that most software companies have testing teams that range from small to large. Meanwhile, only a small percentage of software industries (5.26%) do not have any software testers. The most common number of software testers in the industry is between 1 and 5 individuals, accounting for 35.09%. Furthermore, 29.82% of software companies have between 6 and 25 testers, while 12.28% employ between 26 and 50 testers.

TABLE 6. PROFILE OF RESPONDENTS IN THE SOFTWARE INDUSTRY, NUMBER OF SOFTWARE TESTER

No.	Category	% From Respondent
1	There are no software testers available.	5,26%
2	The number of software testers ranges from 1-5 people	35,09%
3	The number of software testers ranges from 6-25 people	29,82%
4	The number of software testers ranges from 26-50 people	12,28%
5	The number of software testers is 50+ people	17,54%

The survey results comparing the involvement of software developers and software testers reveal a significant disparity. The participation of software developers is generally greater than that of software testers, regardless of the organization's size—be it small, medium, or large. Most software development companies employ only 1 to 5 software testers, and 5.26% of software industries do not employ any software testers at all. This indicates that software testing receives less attention compared to code implementation. However, most

software industries are starting to acknowledge the critical importance of software testing.

The discussion of the survey results is organized into three sections: software testing methodology, software testing tools, and software testing documentation.

4.1 Software Testing Methodology

The first section of the questionnaire discusses the testing strategy in software testing, which is measured based on three parameters: the selected test level, the method of test case development during the design phase, the approach to test execution and performance testing selected during the software testing process. These four parameters were selected because they are related to the general implementation of software testing and are commonly employed in accordance with the testing process standards outlined in ISO/IEC/IEEE 29119 for software testing [4].

Test levels refer to the various types of software testing organized according to the hierarchy of software components [18], [20]. These levels are categorized into four stages: unit testing, integration testing, system testing, and user acceptance testing [18], [20]. Unit testing is a form of software testing that aims to identify defects at the unit level, whether in the unit code during white-box testing or in unit features during black-box testing. Integration testing seeks to uncover errors in component interfaces or data flow when units interact with one another. System testing focuses on detecting defects related to software performance. Lastly, user acceptance testing (UAT) evaluates the software's quality against business requirements, typically conducted by the end user. The survey results regarding the implementation of test levels in the software industry are presented in Table 7.

TABLE 7. IMPLEMENTATION OF TESTING LEVELS IN THE SOFTWARE INDUSTRY

No.	Testing Role	% From Respondent
1	Unit testing	84,21%
2	Integration testing	75,44%
3	System testing	73,68%
4	User acceptance testing	89,47%

Table 7 illustrates the critical importance of testing at all levels within the software industry. This is evidenced by the fact that most software companies conduct testing across every level. UAT is the most emphasized type, highlighting the priority of final validation from the end user's perspective to ensure that business needs are met. Following UAT, unit testing is also prevalent, with a usage rate of 84.21%, as it plays a vital role in maintaining software quality compared to the other two testing levels. Integration testing and system testing are conducted after unit testing is completed and are slightly less common, with usage rates of 75.44% and 73.68%, respectively; however, they remain widely practiced. These two types of testing focus on inter-module interactions and the overall functionality of the entire system.

Test design techniques are utilized to create test cases that guide the software testing process based on specified requirements [21]. A survey on test design techniques categorizes them into three primary approaches: specification-based, structure-based, and experience-based [4]. Specification-based testing, often referred to as black box testing, focuses on functional testing derived from requirement specifications. Structure-based testing, commonly known as white box testing, evaluates the effectiveness of the code structure in executing its functions. In contrast, experience-based testing is designed for experienced testers who establish testing objectives for various test cases. The survey results regarding the application of test design techniques in the software industry are presented in Table 8.

TABLE 8. IMPLEMENTATION OF TEST DESIGN TECHNIQUES IN THE SOFTWARE INDUSTRY

No.	Testing Role	% From Respondent
1	Specification based (black box testing)	87,72%
2	Structure based (white box testing)	54,39%
3	Experience based	77,19%
4	Not conducting test case development	7,02%

The survey results on test design techniques (Table 9) indicate that black box testing is the most employed method, utilized by 87.72% of respondents. In contrast, white box testing is adopted less frequently, with a usage rate of 54.39%; however, it remains a significant component of the testing process. On the other hand, 77.19% of software companies utilize experience-based approaches when designing test cases, relying on the intuition of the tester. This indicates that experience-based testing is still prevalent, highlighting the importance of the tester's expertise in identifying issues that may not be captured by formal testing methods.

Test execution refers to the process of executing test case procedures based on the test design to identify defects and ensure software quality [22]. The survey on test execution is categorized into four techniques: static testing, dynamic testing, test automation, and manual testing [4]. Static testing involves inspecting the source code without running the program, typically conducted by senior developers during code review sessions. In contrast, dynamic testing entails executing the program and is usually performed by independent testers. Automated testing uses engines to execute test cases, while manual testing is carried out manually by testers during the testing process. The survey results regarding the implementation of test execution in the software industry are presented in Table 9.

TABLE 9. IMPLEMENTATION OF TEST EXECUTION IN THE SOFTWARE INDUSTRY

No.	Testing Role	% From Respondent
1	Static testing	43,86%
2	Dynamic testing	85,96%
3	Automation testing	63,16%
4	Manual testing	94,74%

Table 9 illustrates that manual testing continues to dominate test execution, accounting for 94.74% of the total. Dynamic testing is also extensively utilized in the software industry, with a usage rate of 85.96%. On the other hand, automation testing is becoming more popular, currently at 63.16%, although it has not yet fully replaced manual testing. In contrast, static testing is employed less frequently, with a usage rate of 43.86% compared to other testing techniques. Overall, manual testing remains essential for ensuring that software functions as expected. Furthermore, dynamic testing is favored because it

facilitates the validation of functionality and behavior in real operational environments.

Performance testing is measured based on seven parameters outlined in the software testing process standards: functionality testing, load testing, stress testing, volume testing, compatibility testing, interoperability testing, and portability testing [4]. The primary objective of performance testing is to assess the software's overall performance. The specifics of the seven types of performance testing are as follows: (1) Functionality testing focuses on evaluating the software's compliance with requirement specifications. (2) Load testing assesses the software's performance in handling a specified number of concurrent requests. (3) Stress testing examines the software's behavior under conditions that exceed normal usage limits. (4) Volume testing evaluates the software's performance when managing a significant increase in data. (5) Compatibility testing assesses the software's performance across various platforms. (6) Interoperability testing investigates the software's interaction with other systems, whether within the same or different environments. (7) Portability testing ensures that the software can operate effectively in diverse environments. The survey results concerning the implementation of performance testing in the software industry are presented in Table 10.

TABLE 10. IMPLEMENTATION OF PERFORMANCE TESTING IN THE SOFTWARE INDUSTRY

No.	Testing Role	% From Respondent
1	Load testing	71,93%
2	Stress testing	77,19%
3	Volume testing	64,91%
4	Combability testing	75,44%
5	Interoperability testing	64,91%
6	Portability testing	61,40%

Table 10 illustrates that the software industry places significant emphasis on compatibility testing and stress testing, prioritizing the stability of software under various conditions and operational environments. Stress testing is the most frequently conducted performance test, accounting for 77.19% of tests performed, compared to other types. Meanwhile, 75.44% of software companies conduct compatibility testing, 64.91% perform volume testing and interoperability testing, and 61.40% carry out portability testing. The fact that the percentage of all types of performance tests exceeds 60% indicates that the software industry considers it crucial for software to perform effectively under heavy data loads and to maintain stability across different environments.

4.2 Software testing tools

The second topic of discussion from the software testing questionnaire survey results is testing tools. The use of these tools was assessed through questions regarding the selection of tools employed at various testing levels for web and mobile applications. The survey on testing tools is categorized into three areas: performance testing tools, test automation tools, and bug/defect reporting tools. The selection of these three types of testing tools is essential in specific scenarios commonly encountered in the software industry. Performance testing tools are utilized to evaluate the reliability of software, particularly in critical systems [23]. Test automation tools are beneficial during regression testing for software that undergoes numerous code changes [24], [25]. Additionally, test reporting activities involve documenting the results of test case execution during the testing process [26]. Test reporting plays a vital role in tracking defect status, and the use of defect reporting tools facilitates this reporting process [27]. The survey results regarding the use of testing tools in the software industry are presented in Table 11.

TABLE 11. UTILIZATION OF TESTING TOOLS IN THE SOFTWARE INDUSTRY

No.	Testing Tools	% From Respondent
1	Load testing tools	63,16%
2	Test automation tools pada functional testing web application	64,91%
3	Test automation tools pada functional testing mobile application	43,86%
4	API testing tools	66,67%
5	Bug/defect reporting tools	89,47%

Based on Table 11, many software companies have already adopted testing tools during the software testing process. The use of bug reporting tools and API testing is the most common, while test automation in certain categories is slightly less common, particularly in mobile application testing (43.86%). The fact that 89.47% of respondents utilize bug reporting tools highlights the importance of bug reporting and tracking activities, which are essential during the testing life cycle. Meanwhile, 64.91% of respondents employ automation testing tools to test the functionality of web applications, and 63.16% use load testing tools for performance testing.

Tables 12, 13, 14, 15 and 16 illustrate the distribution of testing tools utilized across various types of testing, including load testing, web automation testing, mobile automation testing, and bug reporting. The practical application of these testing tools demonstrates significant diversity; however, certain tools are consistently favored by many professionals in the software industry. In addition to utilizing widely available testing tools, some companies opt for less common alternatives or even develop proprietary tools to meet specific testing requirements. According to survey results, JMeter is the most widely used tool for load testing, while Selenium is the preferred choice for web automation testing. Alongside Selenium, Catalon Studio and Cucumber are also frequently employed for web automation testing in the software industry compared to other tools. Conversely, Postman is highly regarded for API testing. Finally, Jira and spreadsheets are commonly used for documenting bugs throughout the testing lifecycle.

TABLE 12. UTILIZATION OF LOAD TESTING TOOLS IN THE SOFTWARE INDUSTRY

No.	Testing Tools	% From Respondent
1	JMeter	38,60%
2	LoadRunner	5,26%
3	Locust	3,51%
4	Internally Developed Tools	3,51%
5	Application Performance Monitoring (APM)	1,75%
6	K6	1,75%
7	Go Routine	1,75%
8	AWS	1,75%
9	Not Utilizing Testing Tools	45,61%

TABLE 13. UTILIZATION OF WEB AUTOMATION TESTING TOOLS IN THE SOFTWARE INDUSTRY

No.	Testing Tools	% From Respondent
1	Selenium	35,09%
2	Catalon Studio	19,30%
3	Cucumber	8,77%
4	Cypress	3,51%
5	Playwright	3,51%
6	Internally Developed Tools	3,51%
7	Sentry	1,75%
8	RPA (Robot Framework)	1,75%
9	Karma	1,75%
10	Jest	1,75%
11	Itest	1,75%
12	Not Utilizing Testing Tools	38,60%

TABLE 14. UTILIZATION OF MOBILE AUTOMATION TESTING TOOLS IN THE SOFTWARE INDUSTRY

No.	Testing Tools	% From Respondent
1	Appium	8,77%
2	Espresso (for Android)	8,77%
3	Catalon Studio	7,02%
4	XCUITest (for iOS)	5,26%
5	Itest	3,51%
6	Cucumber	1,75%
7	Flutter integration test	1,75%
8	Maestro	1,75%
9	Selendriod	1,75%
10	Robolectric	1,75%
11	UI Automator	1,75%
12	AWS Device Farm	1,75%
13	Not Utilizing Testing Tools	64,91%

TABLE 15. UTILIZATION OF BUG API TESTING TOOLS IN THE SOFTWARE INDUSTRY

No.	Testing Tools	% From Respondent
1	Postman	64,91%
2	SoapUI	12,28%
3	Insomnia	3,51%
4	Swagger	3,51%
5	Not Utilizing Testing Tools	31,58%

TABLE 16. UTILIZATION OF BUG REPORT TOOLS IN THE SOFTWARE INDUSTRY

No.	Testing Tools	% From Respondent
1	Jira	49,19%
2	Spreadsheets	35,19%
3	Trello	15,79%
4	Github	12,28
5	Mantis	7,02%
6	AzureDevops	3,51%
7	Redmine	3,51%
8	Clickup	3,51%
9	BugZilla	3,51%
10	Team Foundation Server	1,75%
11	Ms. Planner	1,75%
12	Microsoft 365	1,75%
13	ServiceNow	1,75%
14	Testrail	1,75%
15	QATouch	1,75%
16	Gitea	1,75%
17	Internally Developed Tools	1,75%
18	Phabricator	1,75%
19	Not Utilizing Testing Tools	8,77%

4.3 Software testing documentation

The third section of the questionnaire discusses test documentation, aiming to understand how many software companies prioritize reporting test results during the software testing process. The survey on test documentation is structured around the various stages of testing, which include activities related to designing test cases and executing test procedures [4]. Consequently, the survey is divided into two categories. First, the test design specification and test procedure specification documents serve as guides for executing tests by representing testing scenarios. Second, the test execution document is utilized to gather information about the software testing results and to report any identified bugs. The survey results regarding the implementation of test documentation in the software industry are presented in Table 17.

TABLE 17. IMPLEMENTATION OF TEST DOCUMENTATION IN THE SOFTWARE INDUSTRY

No.	Testing Role	% From Respondent
1	Test design specification dan test procedure specification documentation	71,93 %
2	Test execution documentation	85,96 %
3	Bug report documentation	89,47 %
4	Not documenting	1,85%

Based on Table 17, most software industries prioritize documentation activities related to bug reporting (89.47%) and test execution (85.96%). These two types of documentation focus on managing and reporting the results identified during testing, as well as ensuring that every step of the testing process is accurately recorded. Meanwhile, 71.93% of the industries create test case designs, indicating that they consider this activity important. However, test case design receives less attention compared to bug reporting and test execution. Conversely, a small percentage (1.85%) of software industries conduct testing without documenting the results. This indicates that most software industries recognize the importance of documenting the testing process to maintain software quality.

5. CONCLUSIONS

The survey research on the implementation of software testing has provided a clear depiction of the testing processes within the Indonesian software industry. The scope of the study includes the approaches to designing testing procedures, executing tests, and documenting software testing results. Based on the discussion in the results and discussion section, the conclusion of the research on the implementation of software testing processes in the Indonesian software industry closely align with the standard processes outlined in ISO/IEC/IEEE 29119 for software testing. This alignment encompasses the software testing methodology, tools, and documentation. Generally, the implementation of the software testing methodology includes test levels, test case development, test case execution, and test result reporting, along with bug reporting. This is evidenced by the following five conditions: (1) most software companies conduct testing at all levels; (2) the specification-based technique for designing test cases is the most preferred compared to the structure-based and experience-based

approaches; (3) manual testing and dynamic testing are the most commonly used methods for test execution; (4) The software industry has integrated both manual and automated testing approaches; however, the implementation of manual testing continues to be prioritized over automated testing and (5) performance testing is widely conducted to ensure software quality. Meanwhile, the implementation of testing tools has been integrated throughout the testing process, encompassing load testing, web testing, mobile testing, API testing, and bug reporting tools. On the other hand, the software industry has extensively documented testing activities, particularly in bug tracking. However, the involvement of software testers in software development has not received as much attention as that of software developers. Several aspects were not addressed in this study that could be explored in future research, including a survey on the utilization of testing tools at the unit level, methodologies for applying black box and white box testing at each unit level, and a more in-depth examination of performance testing for assessing non-functional requirements.

REFERENCES

[1] X. Jia, 'The Role and Importance of Software Testing in Software Quality Management', *Journal of Industry and Engineering Management*, vol. 1, no. 4, pp. 39–44, 2023, doi: https://doi.org/10.62517/jiem.202303406.

[2] M. Bajjouk, M. E. Rana, C. R. Ramachandiran, and S. Chelliah, 'Software testing for reliability and quality improvement', *Journal of Applied Technology and Innovation*, vol. 5, no. 2, pp. 40–46, 2021, Accessed: Sep. 14, 2024. [Online]. Available: https://jati.sites.apiit.edu.my/files/2021/03/Volume5_Issue2_Paper7_2021.pdf

[3] O. Dahiya and K. Solakin, 'A Study on Identification of Issues and Challenges Encountered in Software Testing', in *Proceedings of International Conference on Communication and Artificial Intelligence*, Singapore: Springer, May 2021, pp. 549–556. doi: https://doi.org/10.1007/978-981-33-6546-9_52.

[4] ISO/IEC JTC1/SC7/WG26, 'Test processes', in *ISO/IEC 29119 Software and systems engineering - Software testing*, 2nd ed., IEEE, 2021, ch. 2. doi: https://doi.org/10.1109/IEEESTD.2021.9591508.

[5] S. Kumar, 'Reviewing Software Testing Models and Optimization Techniques: An Analysis of Efficiency and Advancement Needs', *Journal of Computers, Mechanical and Management*, vol. 2, no. 1, pp. 43–55, Feb. 2023, doi: https://doi.org/10.57159/gadl.jcmm.2.1.23041.

[6] S. Ergasheva and A. Kruglov, 'Software Development Life Cycle early phases and quality metrics: A Systematic Literature Review', in *Conference Information Technologies, Telecommunications and Control Systems (ITTCS)*, Innopolis, Russia: IOP Publishing Ltd, Dec. 2020, pp. 1–13. doi: https://doi.org/10.1088/1742-6596/1694/1/012007.

[7] I. Santos, S. M. Melo, P. S. L. Souza, and S. R. S. Souza, 'A survey on the practices of software testing: a look into Brazilian companies', *Journal of Software Engineering Research and Development*, vol. 10, pp. 1–15, May 2022, doi: https://doi.org/10.5753/JSERD.2022.786.

[8] Ralf. Bierig, Stephen. Brown, Edgar. Galván, and Joseph. Timoney, *Essentials of Software Testing*. Cambridge University Press, 2021. doi: https://doi.org/10.1017/9781108974073.

[9] M. Beller, G. Gousios, and A. Zaidman, 'How (Much) Do Developers Test?', in *Proceedings - International Conference on Software Engineering*, IEEE Computer Society, May 2015, pp. 559–562. doi: https://doi.org/10.1109/ICSE.2015.193.

[10] V. Garousi, M. Felderer, M. Kuhrmann, K. Herkiloğlu, and S. Eldh, 'Exploring the industry's challenges in software testing: An empirical study', *Journal of Software: Evolution and Process*, vol. 32, no. 8, pp. 1–23, Aug. 2020, doi: https://doi.org/10.1002/smr.2251.

[11] P. Waychal, L. F. Capretz, J. Jia, D. Varona, and Y. Lizama, 'Practitioners' Testimonials about Software Testing', in *Proceedings - 2021 IEEE International Conference on Software Analysis, Evolution and Reengineering, SANER 2021*, Institute of Electrical and Electronics Engineers Inc., Mar. 2021, pp. 582–589. doi: https://doi.org/10.1109/SANER50967.2021.00070.

[12] Hariyanto, T. Dirgahayu, and H. P. P, 'Software Quality Assurance pada Perusahaan Pengembang Perangkat Lunak Skala Kecil dan Menengah', *Jurnal Riset Teknologi dan Inovasi Pendidikan (Jartika)*, vol. 3, pp. 115–130, Jun. 2020, doi: http://dx.doi.org/10.36765/jartika.v3i2.265.

[13] T. Wahyuningrum and K. Mustofa, 'A Systematic Mapping review of Software Quality Measurement: Research Trends, Model, and Method', *International Journal of Electrical and Computer Engineering*, vol. 7, no. 5, pp. 2847–2854, Oct. 2017, doi: https://doi.org/10.11591/IJECE.V7I5.PP2847-2854.

[14] Y. Wang, M. V. Mäntylä, S. Demeyer, K. Wiklund, S. Eldh, and T. Kairi, 'Software Test Automation Maturity -- A Survey of the State of the Practice', in *In Proceedings of the 15th International Conference on Software Technologies – ICSOFT,* SciTePress, Jul. 2020. doi: https://doi.org/10.5220/0009766800270038.

[15] T. Hynninen, J. Kasurinen, A. Knutas, and O. Taipale, 'Software Testing: Survey of the Industry Practices', in *International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, IEEE, May 2018, pp. 1449–1454. doi: https://doi.org/10.23919/MIPRO.2018.8400261.

[16] J. Ponto, 'Understanding and Evaluating Survey Research', *J Adv Pract Oncol*, vol. 6, no. 2, pp. 168–171, Apr. 2015, doi: https://doi.org/10.6004/jadpro.2015.6.2.9.

[17] S. Izzat and N. N. Saleem, 'Software Testing Techniques and Tools: A Review', *Journal of Education and Science*, vol. 32, no. 2, pp. 30–40, Jun. 2023, doi: https://doi.org/10.33899/edusj.2023.137480.1305.

[18] D. S. Taley and B. Pathak, 'Comprehensive Study of Software Testing Techniques and Strategies: A Review', *International Journal of Engineering Research and Technology (IJERT)*, vol. 9, no. 8, pp. 40–46, Aug. 2020, doi: https://doi.org/10.17577/ijertv9is080373.

[19] S. Reid, 'The Personal Test Maturity Matrix', in *Japan Symposium on Software Testing (JaSST)*, Mar. 2014. Accessed: Sep. 09, 2024. [Online]. Available: https://jasst.jp/symposium/jasst14tokyo/report.html

[20] S. He and P. Carracedo, 'Software Testing Summary for Chemoinfortics', in *MOL2NET'21, Conference on Molecular, Biomedical & Computational Sciences and Engineering*, Sciforum, Oct. 2021, pp. 1–4. doi: https://doi.org/10.3390/mol2net-07-11214.

[21] Kusum, P. Talwar, A. Puri, and G. Kumar, 'Overview of software testing', *Global Journal of Engineering and Technology Advances*, vol. 19, no. 1, pp. 104–112, Apr. 2024, doi: https://doi.org/10.30574/gjeta.2024.19.1.0060.

[22] K. S. Thant and H. H. K. Tin, 'The Impact Of Manual And Automatic Testing On Software Testing Efficiency And Effectiveness', *Indian Journal of Science and Research*, vol. 3, no. 3, pp. 88–93, Sep. 2023, Accessed: Sep. 14, 2024. [Online]. Available: https://www.ijsronline.org/issue/20230714-032703.942.pdf

[23] S. Pargaonkar, 'A Comprehensive Review of Performance Testing Methodologies and Best Practices: Software Quality Engineering', *International Journal of Science and Research (IJSR)*, vol. 12, no. 8, pp. 2008–2014, Aug. 2023, doi: https://doi.org/10.21275/sr23822111402.

[24] V. Biju and S. Ali, 'Automation of Purchase Order in Microsoft Dynamics 365 by Deploying Selenium', in *8th International Conference of Security, Privacy and Trust Management (SPTM 2020),* Helsinki, Finland: Computer Science and Information Technology, Jun. 2020, pp. 101–114. doi: https://doi.org/10.5121/csit.2020.100610.

[25] S. K. Alferidah and S. Ahmed, 'Automated Software Testing Tools', in *2020 International Conference on Computing and Information Technology, ICCIT 2020*, Tabuk, Saudi Arabia: Institute of Electrical and Electronics Engineers Inc., Sep. 2020, pp. 183–86. doi: 10.1109/ICCIT-144147971.2020.9213735.

[26] ISO/IEC JTC1/SC7/WG26, 'Test documentation', in *ISO/IEC 29119 Software and systems engineering - Software testing*, 2nd ed., IEEE, 2021, ch. 3. doi: https://doi.org/10.1109/IEEESTD.2021.9591577.

[27] H. Singh, 'Documenting Test Results', in *Structural Materials*, 1st ed., Singapore: Springer, 2021, ch. 5, pp. 147–162. doi: https://doi.org/10.1007/978-981-16-3211-2_5.

AUTHORS

Asri Maspupah
Department of Computer and Informatic Engineering, Politeknik Negeri Bandung, with expertise in software testing and software engineering.

Ani Rahmani
Department of Computer and Informatic Engineering, Politeknik Negeri Bandung, with expertise in software engineering and software testing.

Joe Lian Min
Department of Computer and Informatic Engineering, Politeknik Negeri Bandung, with expertise in software engineering.

Trisna Ari Roshinta
Computer Engineering, Budapest University of Technology and Economics, Hungaria, with expertise in intellegent system.