Testing Of Industrial Practice Monitoring Information Systems Using The Black Box Method (Case Study: Informatics Department Of Surabaya State University)

Ronggo Alit
Departement Of Informatic Engineering
State University Of Surabaya
Surabaya, Indonesia
ronggoalit@unesa.ac.id

I Kadek Dwi Nuryana
Departement Of Informatic Engineering
State University Of Surabaya
Surabaya, Indonesia
kadekdwinuryana@gmail.com

Aries Dwi Indriyanti
Departement Of Informatic Engineering
State University Of Surabaya
Surabaya, Indonesia
ariesdwi@unesa.ac.id

Abstract*— Industrial Practice is a means for students to understand how to gain knowledge from the web-based Industrial Practice Monitoring Information System (SEMPI), which is a system that can provide information about industrial practice programs online. This system has an advantage in terms of the speed of presentation of the resulting information. In addition, this system is web-based so that it can be accessed at any time. Measuring the quality of current software is very necessary before a software will be used by users. This aims to find out the weaknesses of the system so that the resulting data can match the data entered after the data is executed. The technique of this research is the incoming and outgoing mail information system then carried out using Black Box testing. This method focuses on data entry, system display, memory usage and speed of data execution so that if the input data does not match what is expected, the system will fail. The results of testing using this method are said to be very good because all weaknesses in the system can be identified before use.

Keywords*— Information systems, Industrial Practices, monitoring

I. INTRODUCTION

The development of information technology also touches the processes within the scope of the university in order to increase work effectiveness and efficiency. One example is the academic performance of the archives section which is less efficient because it still applies a conventional filing system. Industrial Practice is a means for students to understand how the knowledge gained during lectures can be applied in industry or institutions. In addition, students are also expected to be able to analyze the system in finding alternative problem-solving processes to make them more efficient and as a first step to form a work ethic and professionalism before engaging in the world of work. Industrial practice is also an understanding and introduction to students about conditions in the field that must be known. The process of applying for and implementing industrial practices in an industry or company is certainly not without problems. The problem that arises is that there is no cooperation between departments and industry so that students must find their own industries that accept students for industrial practice. Even if the intended industry accepts students for industrial practice, the implementation time determined by the industry is sometimes not in line with student expectations. Even during the implementation of industrial practices, there was no industrial practice program approved by the industry and departments. Meanwhile, to get maximum results in implementing industrial practices, programs for students during industrial practices must be coordinated between universities and industry (Polat et al., 2010, Mircea & Ana-andreea, 2013).

All programs during industrial practice were fully handed over to the industry so that the monitoring and assessment process by the supervising lecturer was not effective which ultimately followed the values given by the company supervisor. The large number of students and the small number of supervisors is also a separate problem (Koc et al. 2014). Furthermore, the suitability of students with mentors from industry must be considered. If not, students will feel uncomfortable in carrying out industrial practices which in the end the achievement of student competence is not optimal (Sweitzer & A. King, 2013). Students should try to get to know supervisors from industry more closely so that they feel comfortable while working. From the background above, it is necessary to have a synergistic program regarding industrial practice procedures, the role of supervising lecturers, the role of supervisors from industry and student jobdesks during industrial practice to support the achievement of the objectives of the industrial practice itself. Software testing that is carried

out imperfectly will certainly have an adverse effect on the quality of the software produced. Ineffective and incomplete software testing can lead to various problems when the software is used by end-users. Automatically can increase the efficiency of the testing process to identify parts of the software that are prone to failure (Catelani 2011).

In the use of information systems, there are often several problems experienced by users, such as system bugs. To find these errors, it is necessary to test the system (Jaya, 2018). Software testing is an important stage in the software development life cycle (SLDC) which aims to ensure the quality and functionality of the software being developed (Hidayat & Muttaqin, 2018). By conducting testing, the development team can save time and money during the system development process (Pratala et al., 2020). Three testing techniques are commonly used by software testers, namely black-box testing (testing software functionality), white-box testing (testing system code), and experience-based techniques (tester experience) (Ikhlaashi & Putro, 2019). Black-box testing is the most widely used testing technique because it only tests the system based on its functionality (Parlika et al., 2020). In the black-box testing technique, there are several methods, namely equivalence partitioning, boundary value analysis, decision tables, state transition testing, and use case testing (Ikhlaashi & Putro, 2019).

In software testing there are various methods that can be used for testing, for example the Black Box Testing method. Generally, a software testing method only covers an area in the software (Data Driven Testing) only tests the ability of the software to receive and process input data. Of course, a software testing system must be able to test various aspects of the software so that the use of more than one type of sub-test is highly desirable (Ghalin 2014). Based on the method used, it will be known the weaknesses in the information system after testing using the Black Box method and how to find out which output is considered valid (according to the needs of kermawa). which is entered after the data is executed and avoids deficiencies and errors in the application before being used by the user.

Black box testing or more commonly known as functional testing is a software testing method used to test software without knowing the internal structure of the code or program. In this test, the tester is aware of what the program has to do but has no knowledge of how to do it. In Black Box Testing, testing is carried out based on application details such as the appearance of the application, the functions available in the application, and the suitability of the function flow with the business processes desired by the customer. Black-box testing is more testing the outer appearance (Interface) of an application so that it is easy for users to use. This test does not see and test the program's source code. Black-box testing works by ignoring the control structure so that its attention is only focused on domain information.

The “Black Box” test simply consists of reviewing the functionalities of the app, i.e. if it does what it is supposed to do, no matter how it is done. Its internal structure and function are not studied. Thus the tester needs to know what the role of the system is, and its functions, but not know its internal mechanisms. He has a “user” profile.

II. METHOD

Black Box Method System testing aims to see whether the system that has been made is in accordance with the original purpose of manufacture and is feasible to use. Testing on the system uses the Black Box method, the aim is to find out that the parts in the application system correctly display error messages if an error occurs in data input (Sandy 2015). Black Box Testing itself is a test that is carried out only by observing the results of execution through test data and checking the functionality of the software. This black box testing focuses on system functions (Rizki 2015).

Here are 10 test types of methods Black Box according to (Julian 2015):

1. Equivalence Partitioning : Splits input into data classes that can be used for generate test cases.
2. Boundary Value Analysis / Limit Testing: Allows selecting test cases that test input value constraints. This is a complement to Equivalence Partitioning.
3. Comparison Testing: Test each version with the same data to make sure they all produce the same output.
4. Sample testing: Involves several selected values from an equivalent class.
5. Robustness Testing : The input data is selected outside the specifications that have been defined. The purpose of this test is to prove that there is no error if the input is invalid
6. Behavior Testing: Test results cannot be evaluated if you only do the test once, but you can evaluated if the test is performed several times, for example on a test stack data structure.
7. Performance Testing : Evaluating the program's ability to operate correctly is viewed From a demand perspective, for example: data flow, memory usage size, execution speed.
8. Requirements Testing: Requirements specifications associated with the software are identified at the requirements specification and design stages.
9. Endurance Testing : Involves test cases that are repeated a certain number of times.
10. Cause – Effect Relationship Testing : Divide the requirements specification into parts that have work possibilities

Trial Scenario Testing using random data input aims to ensure the system refuses to store input data in the database, so that the system is said to be feasible to use. Here's how to test each type of the Black Box method:

1. Equivalence Partitioning This test is carried out on a form that already exists in the mail information system Sign in and out by entering

data that do not match the data type or enter random data.

2. Boundary Value Analysis This test is to ensure that input data that exceeds a predetermined limit cannot be stored properly in the database.
3. Comparison Testing Comparing the display interfaces system on different web browsers
4. Sample testing This test is to ensure the selected value can produce good data and is in accordance with the input data from the user.
5. Robustness Testing The tester will enter random data to prove that there is no error if the input is not valid.
6. Behavior Testing This test is done by creating new data many times to avoid stack data.
7. Performance Testing This test evaluates the program's ability to operate correctly from the point of view of memory usage,
8. Requirements Testing This type only looks at the requirements specifications of the system from the manufacturing system to testing.
9. Endurance Testing This type is to ensure whether the results of mathematical operations on this system are correct or wrong.
10. Cause – Effect Relationship Testing Testing involving input conditions and data flow starting from Input, View, Update, Delete and Search.

So that from the method of testing above that has been explained to carry out tests of several types of tests which will later be implemented on the information system, the results can be seen in a table. It is hoped that there will be no input errors during the test to ensure the system produces good data.

III. RESULT AND DISCUSSION

A. Implementation and trials

Implementation and trials were carried out on an industrial practice monitoring information system with the aim of finding out the deficiencies that exist in the system before the system is used by users.

B. Valid Data Testing

Testing using the correct data by non-IT students in the hope that the system is also able to receive data without any good errors. the goal is Enter data correctly for example the system accepts input data to be stored in the database and Enter data in the form of random data to ensure the system refuses to store input data in database.

In the test results there is a test case table that functions to conclude whether the system is successful in testing this type or not. Following are some explanations of the test table to be used: Input is an explanation of entering correct data or random data, Expected results are the results that must occur when in the testing process, Output is the result of testing after the system has been tested, Conclusion is a conclusion whether the system is declared successful or failed.

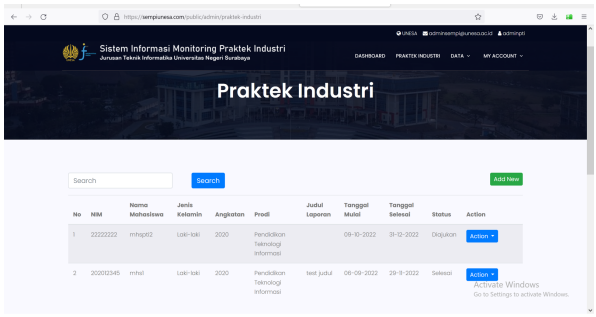

Fig. 1. SEMPI Application Data Input

In Figure 1 is the process of testing the correct data by entering the data needed to monitor industrial practices.

TABLE I. CORRECT DATA TEST RESULTS

Input	*Results Hope*	*Output*	Conclusion
Fill in the data on mandatory information	The system accepts all data	The system accepts all data enter with the appropriate data type	[√] Success [ ] Fail

based on table 1 above the results of the Correct Data Test on the industrial practice form show success and no errors

C. Invalid Data Testing

Random Data Testing (Invalid Data) Testing using random data input aims to ensure the system refuses to store input data in the database, so that the system is said to be feasible to use. Testing will be carried out with 10 types of Black Box methods:

1. Equivalence Partitioning

This test was carried out on the form already exists in the SEMPI information system by entering data that does not match the data type or entering random data.

TABLE I. EQUIVALENCE PARTITIONING TEST RESULTS

Input	*Results Hope*	*Output*	Conclusion
Input : *ID* : 911 *Password* 1542376 (Random)	The system refuses to save the input data	The system returns to the main menu and cannot enter the SEMPI information system	[√] Success [ ] Fail

Insert input with html text	System refused to save because of the data in the form of html text and the system does not experience interfaces errors	System able to store data in databases and change the appearance of the system	[ ] Success [√] Fail

2. Boundary Value Analysis

This test is to ensure that data entry that exceeds a predetermined limit cannot be stored properly in the database, and the system only displays data that is less than the data limit.

TABLE II. BOUNDARY VALUE ANALYSIS TEST RESULTS

Input	*Results Hope*	*Output*	Conclusion
Fill in the information form must exceed the data limit	The system stores data and displays data according to the limit	The system displays the appropriate data with limits data	[√] Success [ ] Fail

3. Comparison Testing

This type will not look for errors in the input, but instead differentiates the appearance of the system interfaces on different web browsers, this type is only for software redundancy and to test each version with the same data to ensure all versions produce the same output

TABLE III. COMPARISON TESTING TEST RESULTS

Input	*Results Hope*	Export results of irregular	*Internet Explorere*
Appearance save as PDF / PNG	Export results of irregular incoming mail data	[ ] Success [√] Fail	[ ] Success [√] Fail

4. Sample Testing

This test is to ensure the selected value can produce good data and is in accordance with the input data from the user.

TABLE IV. SAMPLE TESTING RESULTS

Input	*Results Hope*	*Output*	Conclusion
input the data search to ensure value selected	The system managed to find the data	System worked displays related data	[√] Success [ ] Fail

5. Robustness Testing

Random data testing where the tester will enter random data to prove that there is no error if the input is invalid. When the system displays invalid output results, the system is not said to have failed in this test.

TABLE V. ROBUSTNESS TESTING RESULTS

Input	*Results Hope*	*Output*	Conclusion
Edit student data with data symbol * Becomes #	The system accepts additional data Student	System worked accept addition of data with random data	[√] Success [ ] Fail

6. Behavior Testing

This test is carried out by creating new data many times to avoid data stacks and the system can accept data with an amount of more than 50.

TABLE VI. BEHAVIOR TESTING RESULTS

Input	*Results Hope*	*Output*	Conclusion
Show results of adding data users in a manner repeatedly becomes #	The system managed to store data and did not experience a data stack	The system saved successfully data as many as more than 50 and the system does not experience problems	[√] Success [ ] Fail

7. Performance Testing

This test evaluates the program's ability to operate correctly in terms of memory usage flow, data flow and execution speed. The memory usage test was carried out on a similar web browser.

TABLE VII. PERFORMANCE TESTING RESULTS

Input	*Results Hope*	*Output*	Conclusion
Search with the word "Lecturer "	The system displays all letter data related to important words	The system displays successfully mail data sought	[√] Success [ ] Fail

8. Requirement Testing

This type is not called testing because this type only looks at the specification requirements of SEMPI starting from the manufacturing system to testing. The software and hardware requirements used will be explained in the table for this type

TABLEVIII. REQUIREMENT TESTING RESULTS

Hardware	**Database**	***Programming Languange***	***Testing Method***
Prosesor Intel inside core i3 **Memory** 4 GB	*Mysql*	*Php*	*Blax Box Testing* with 10 type of testing

9. Endurance Testing

This type ensures whether the results of mathematical operations in the SEMPI system are correct or incorrect, judging from each menu or sub-menu that contains the number or results of all letters

TABLE IX. ENDURANCE TESTING RESULTS

Input	*Results Hope*	*Output*	Conclusion
The total number of systems same with manual results	The result of the sum is the same, namely 10 student registrants	The result of the sum is equal to that result expected	[√] Success [ ] Fail

10. Cause-Effect Relationship Testing

Tests that involve input conditions and data flow starting from add new, edit, delete and search.

TABLE X. CAUSE-EFFECT RELATIONSHIP TESTING RESULTS

Input	*Results Hope*	*Output*	Conclusion
Delete data with @#. %%/*# @/#@!%/% %/*&%$	The system has successfully deleted the data and a notification appears "data successfully deleted"	The system spawns notification and number data letter @#. %%/*#@ /#@!%/% %/* &%$ is not deleted	[ ] Success [√] Fail

IV. CONCLUSION

After discussing and testing the incoming and outgoing mail information system, the following conclusions are drawn:

1. This test can help find errors in information systems that have been made before the information system is used.
2. Make it easier for a system developer to develop errors that have been found in testing.
3. This information system uses the Black Box method with 10 types of testing with good results, so that some errors or weaknesses in the information system can be found.

REFERENCES

[1]. Andika, Radenal. "Penerapan CI Dalam Pengembangan Sistem Informasi Manajemen Surat dan Pengarsipan" Studi Kasus : PT. Semen Padang. 2011

[2]. Ferdinandus, Sandy. Perancangan Aplikasi Surat Masuk Dan Surat Keluar Pada PT. PLN (Persero) Wilayah Suluttenggo. 2015.

[3]. Junidar. Perancangan informasi arsip surat menyurat di universitas u'budiyah indonesia. 2012.

[4]. Kurniawan, Yoga. "Rancang Bangun Perangkat Lunak Untuk Workflow Pengelolaan Surat Menyurat Dinas Bagian Surat Masuk di Kabupaten Buton Utara". 2012.

[5]. Rouf, Abdul. Pengujian perangkat lunak dengan menggunakan metode white box dan black box. 2012.

[6]. Santoso, Arum Tungga Dewi. Sistem informasi administrasi surat masuk dan surat

keluar pada badan kepegawaian daerah kota semarang. 2014.
[7]. Sutono, D. Sistem Informasi manajemen. 2007.
[8]. Supardi, Julian. Materi Kuliah Black-Box Testing. 2015.
[9]. Syaban, Rizki, Maulana Syaban, H.Bunyamin. "Pengembangan sistem informasi pengelolaan surat masuk dan keluar berbasis web di dinas sosial tenaga kerja dan transmigrasi kabupaten garut menggunakan framework php". 2015.
[10]. Zulkifli. Model Prediksi Berbasis Neural Network untuk Pengujian Perangkat Lunak Metode Black-Box. "Seminar Nasional Aplikasi Teknologi Informasi (SNATI)".2013.
[11]. Hidayat, T., & Muttaqin, M. Pengujian Sistem Informasi Pendaftaran dan Pembayaran Wisuda Online menggunakan Black Box Testing dengan Metode Equivalence Partitioning dan BoundaryValue Analysis. Jurnal Teknik Informatika UNIS JUTIS, 6(1), 2252±5351.2018
[12]. Ikhlaashi, S., & Putro, H. P. Komparasi Dua Teknik Black Box Testing: Equivalence Partitioning dan Boundary Value Analysis. Annual Research Seminar (ARS), 5(1), 8.2019
[13]. Jaya, T. S. Pengujian Aplikasi Dengan Metode Blackbox Testing Boundary Value Analysis (Studi Kasus: Kantor Digital Politeknik Negeri Lampung). Jurnal Informatika: Jurnal Pengembangan IT (JPIT), 3(2), 45±48. 2018
[14]. Parlika, R., Nisaa, T. A., Ningrum, S. M., & Haque, B. A. Studi Literatur Kekurangan dan Kelebihan Pengujian Black Box. 2020
[15]. Pratala, C. T., Asyer, E. M., Prayudi, I., & Saifudin, A. Pengujian White Box pada Aplikasi Cash Flow Berbasis Android Menggunakan Teknik Basis Path. Jurnal Informatika Universitas Pamulang, 5(2), 111. 2020