COMPARISON BETWEEN TRIAL ERROR METHOD AND INTEGRATION TESTING METHOD ON THE SERVICE MENU AT PT. HARTONO ISTANA TEKNOLOGI

[1]Nickolas Hadi Andara Putra, [2]Rosita Herawati
[1,2]Program Studi Teknik Informatika Fakultas Ilmu Komputer,
Universitas Katolik Soegijapranata
[2]rosita@unika.ac.id

ABSTRACT

Here are obstacles in the process of making software at PT. Hartono Technology Palace. At that time, every time a test was carried out on the software being developed, the company would find different results. This causes ineffective and inefficient software development processes. Problems at PT. Hartono Istana Teknologi was found because of the less use of standard software testing methods. That is why the test results are constantly changing, and the software development process cannot be completed quickly. This problem can be solved by applying the software testing method. This research will try to solve the problem by using 2 methods, trial and error, and integration. These two methods will be compared to determine which method is more effective in the software testing process at PT. Hartono Technology Palace. Several test scenarios are also designed to be applied in this study. As a final result, the trial and error method is the most effective method to be applied at PT. Hartono Istana Teknologi. The trial and error method is considered effective because the testing process is more thorough and in accordance with the established concept. While in the integration method, the testing process only focuses on testing data without paying attention to existing functions

Keywords: Software Testing, Trial and Error, Effectiveness

INTRODUCTION

This research was conducted because there was a little confusion in choosing the software testing methods available at PT. Hartono Istana Teknologi. In the beginning, there was no special test method for software testing. It is hoped that there will be a special method for testing software so that it is not difficult to test software. The methods used in this project are Trial and Error and Integration testing. These methods are used because most companies out there also use this method. Then take this method to test the existing software at PT. Hartono Istana Teknologi.

The Trial and Error method itself is a method to achieve a goal through various means. This effort is done several times until finally getting the most suitable way. Errors or errors are recorded for evaluation and as learning material. The effort was carried out in more than one way until it succeeded. Whereas integration testing is a phase in software testing where individual software is combined and tested as a group. Integration testing is carried out to evaluate the compliance of a system or component with the specified functional requirements. It happens after unit testing and before validation testing. Integration testing takes the tested input. Implements the tests specified in the test plan and provides an integrated system output ready for further testing.

The problem-solving solution offered in this research is to use the 2 methods described previously, namely Trial and Error and Integration testing. After the test is successful or has been completed, this study indirectly also compares the two methods. Where from the two methods will look for more effective results for testing the existing software at PT. Hartono Istana Teknologi. The problems that can be obtained are as follows:

1. How are the results made using the Trial and Error method?
2. How are the results made with the Integration testing method?
3. Are there any differences in the results of the two methods?

In this study, the researcher must know and be able to explain the scope and limitations of the research itself. the limitations are as follows:

1. Perform software testing using trial and error methods and integration testing in the service department, service visits, price masters, damage master diagrams, and regional coverage registration.

2. Looking for the most appropriate method for software testing at PT. Hartono Istana Teknologi.

The purpose of this study is to find out the results of the comparison of 2 software testing methods. The two methods are trial and error and integration testing. This research will determine the most appropriate software testing method used for software testing activities.

LITERATURE STUDY

In this chapter I try to describe and describe some appropriate literature studies and related to the project I am working on. It discusses all the journals arranged in a narrative manner with certain information. Divo [1] has shown Coal produced by CV. Bara Mitra Kencana consists of various types based on their calorific value from different pits. This coal calorie is influenced by the geological conditions of the mining location, including coal quality, medium quality, and quality (low quality). The results of calculations using the Trial and Error Method using four parameters of coal quality and consumer demand. Calculations using the Trial and Error Method are close to consumer demand (can be fulfilled). In this study, to analyze the mixing of coal of different qualities so that the comparison between high quality and low quality coal is in accordance with consumer demand. The type of research used is applied research, research is more on the application of science, application, or the use of science for and in society, or for certain purposes (industry, business, etc.). With the results of calculations using the Trial and Error Method from the researchers, there is a GCV of 6297 Cal/gr which is better than the results of the blending analysis of 6000 Cal/gr and can approach consumer demand with a GCV of 6300 Cal/gr.

P.Duval and Lomperski [2] proved that the trial and error method was suitable to be applied to the evolution of control systems. Here refers to the particle accelerator control system and

machine size, the control system, although complex, will not cause damage in the event of a failure. Here is described there are some basic algorithms of the trial and error method that is determine what is the solution to our problem, Try something, Check to see if the problem has been solved, If not: Modify something to a more promising direction and repeat step 3 Or, if the problem is solved then stop. The importance of trial and error is to make mistakes and learn from them. In advancing the capabilities of control systems will be explored, particularly in relation to decision making and mitigation. The result of this method is that it can test multiple solutions for the same thing.

Sekarsari and Tata [3] stated that this study presents an overview of the performance of the PID control technique applied to a Brushless DC motor to regulate speed using an Arduino microcontroller and LabView software. The brushless DC motor was chosen because it has high efficiency both in performance and longer durability. The technique of setting parameters for Kp, Ti and Td uses the Trial and error method. Namely the use of trial and error method for the manufacture and testing of PID controls. The first thing to do is make a system design from the PID control itself. Then there is an overview of the stages of the trial implementation process. In testing, test scenarios are created in the form of a test case table. In the test scenario there are 2 trials, namely trials without PID control and with PID control. Then do the test according to the scenario that has been made. The results obtained in this trial show a small error. Where in the results of the two results are almost the same and the difference is only slightly. The results of the initial calculation (without PID control) obtained the value of Kp = 0.039 Ti = 0.008 and Td = 0.002. When the test is applied to the setpoint value of 1000 rpm, the result obtained is an error of 62.5 when it rises and reaches the setpoint value in a fairly large time. From the research results, it is known that the parameter adjustment technique of Kp, Ti and Td with the Trial and error method produces parameter values of Kp = 0.050, Ti = 0.001 and Td = 0.001. This parameter has the ability to test a calibrated setpoint value with a maximum motor driver voltage of 0 to 1275 rpm with a speed range of 50 rpm. The fastest setting time value ranges from 0 to 11 ms and has the smallest error value difference in steady state conditions.

Tomoki Imai et al. [4]stated that the application of the trial and error method was used in the pilot program, namely Shiranui. Shiranui consists of a user interface, an interpreter and a server that connects the user interface with the interpreter. Shiranui owns a live programming environment similar to YinYang and Apple Swift. The user interface is implemented as an Emacs plugin (900 LoC). The translator and server are developed in C++ (7,200 LoC). Emacs sends an event to the server whenever the user presses a button. The interpreter evaluates the program and the server sends the results back to Emacs. The live programming environment itself is a powerful experimental tool that allows programmers to write programs on trial and error thanks to its fast feedback. Because rapid feedback can include intermediate data such as control flow. One drawback is that this test is temporary.

Florentin and Mike [5] stated that although much research has been done in the area of formal methods and their practical use for the specification and verification of software systems, testing

issues are rarely mentioned by those in the formal methods community. Almost all of the methods currently used to test software are based on experience rather than based on theoretical methods. The formal method X-machine uses, is a mix of finite state machines, data structures and pros. The X-engine incorporates the ability to model data structures. The test set generated by this method has a manageable size like the application process. If the processing function can be computed by several algorithms, then the process of generating test sets can be automated. applied to several case studies and the results are encouraging. Obviously, this method must be supported by automated systems and appropriate tools that do not yet exist.

Rehman et al. [6]states This paper discusses various problems that can arise in component testing by component users at the stage of their integration in the target system. In this survey, a test case was made to facilitate testing and a flowchart was also made. System testing does not show the main concepts and is carried out by the component user when all the various components are integrated and the whole system is ready for use. System testing includes load and performance testing, to test the overall system functionality. This testing system can also be used in the testing phase of a software system. As well as testing the relationship or integration of each menu, button, command and display in it. Therefore, in terms of testing there is also a test flow to find out which one will be tested.

Ursula and Monika [7] stated that one of the main problems of testing is testing data quality. To evaluate the quality, we can use the size coverage test that we get from the test data selection criteria. When we test large programs, we distinguish between module testing and integration testing. During module testing, we concentrate on testing the algorithm and its data, while during integration testing, we concentrate on testing the relationships and interfaces between modules. There are several tests such as effectiveness testing, criteria-based relationship testing on the call graph, criteria interface testing based on data flow. The advantage of the differentiation and integration test module is that we can use robust test data selection criteria, such as an all-loop or all-path iterative free path. In later testing there are some things that are the same. Such as testing algorithms and data. In this case it will be very important because this problem is in component checking. So the existing data must be valid and must be appropriate. And there is also a test between one menu with another menu.

W.T.Tsai et al. [8] states this paper proposes an approach to designing End-to-End (E2E) initiation tests, including specification of test scenarios, creation of test cases, and tool support. integration testing is always a challenge especially if the system under test is large with many subsystems and interfaces. There are several tests, namely the creation of test scenarios and the creation of end-to-end test cases. For the results themselves in the form of a flowchart. The flowchart presentation tier contains users, then there is the middle tier which contains software testing, and the data tier is database testing. This test can be used in terms of creating scenarios for testing. Given that in a test there must be a flow or test scenario. Later there will be some things that are the same at the testing stage, namely in terms of testing the relationship between menus

until the results that enter a menu and the results that are already displayed on the menu must be the same as those in the database.

S Phani Shashank et al. [9]stated that this study aims to find the challenges of testing and integration that exist in Component-based software engineering (CBSE). The question in this research is a survey of the existing integration testing techniques for CBSE and comparing the existing techniques and find the scope to address the existing problems. A systematic survey investigates existing integration testing techniques, understanding the behavior of components and their interactions. As for component testing, testing at run time and the approach to generating test cases for developing components is useful. However, the non-functional aspects of a system consisting of components that can be analyzed and tested have great potential. This must also be considered in testing the software system to be tested. Where in a test not only one technique is used but there are several techniques used. In testing using the integration technique alone is good enough in testing the software system being tested. Because it has caused some problems in the software system.

Youngchul Kim and Robert Carlson [10] stated that this paper focuses on the software testing metrics used in the creation of object-oriented test plans as part of that methodology. The use case action matrix contains a collection of related scenarios that each describes a specific variant of the executable use case action sequence. Software testing is used to increase the productivity of the testing process through priority scenarios. There are 2 scenarios created, namely the scenario of purchasing goods and scenarios of returning the goods purchased. The algorithm for integration testing itself includes the design, development, testing, and maintenance processes. So that the testers can easily test the software system, use cases are made, to make testing easier and there is also the creation of test scenarios. The approach described in this paper emphasizes the creation of a test plan. The biggest benefit is increasing the productivity of process testing.

RESEARCH METHODOLOGY

Analysis

This research uses several problem solving methods. One of them is the Trial and Error testing method. This method is used to perform Software Testing of programs that have been created. The Trial and Error testing method itself is a basic method of solving problems that exist in a program. In general, this method is used to find a solution to a problem in the program that has been created. For problems or errors that exist will be recorded and evaluated to be a learning material. This can be useful for developers because it can help to find problems that exist in the program or software that is made.

Currently, the use of the trial and error testing method is quite attractive to many large companies. As an example, the problem taken in software testing at PT. Hartono Istana Teknologi. Before the new software is launch, it must be tested to known where any error existed in the new software. After testing the software using the trial and error method, the errors found will be recorded. After the testing is completed. the recording results will be given to the developer. The

next step is to wait for the results of the software program improvements that have been tested from the developer. Upon completion, the developer will provide the corrected software program for retesting. The next step is the same as the first step, namely re-testing the software on the corrected program. To assess whether the program is feasible or ready to run, it can be judged according to the concept or design that has been made and also in accordance with the specified validation.

The second test method is Integration Testing. Integration Testing itself is a testing method to combine units/components of the application or software being tested. Integration testing is used after unit testing is complete and before system testing is performed. Testing with the Integration testing method can be used on software that has many menus. as in PT. Hartono Technology Palace. This test is tried to be applied to existing software programs, to find several interconnected menus.

There is a test case as the testing phase of this software. The test case itself is a design in the form of actions taken by the user to verify certain features or functions of a software. The testing actions are carried out as follows.

Tabel 1.   Test Case

| No	Deskripsi Pengujian	Hasil yang diharapkan
01	Pencarian menu dengan search bar	Bisa memunculkan hasil sesuai dengan code yang dimasukkan
02	Saat klik tombol Filter Data maka muncul menu input Filter Data	Bisa memunculkan menu input Filter Data
03	Saat klik tombol search maka akan mencari hal yang diinginkan	Bisa memunculkan hasil sesuai dengan inputan
04	Saat klik tombol Search help maka akan muncul menu tampilan data	Bisa menampilkan daftar list data
05	Saat klik tombol Dropdown akan membuka beberapa pilihan	Bisa menampilkan beberapa pilihan dengan format kebawah
06	Saat mengisi Date akan membuka tampilan kalender	Bisa menampilkan dan memilih tgl/bulan/tahun yang diinginkan
07	Saat klik tombol Baru akan membuka menu untuk create data baru	Bisa menampilkan dan mengisi menu create data baru

No	Deskripsi Pengujian	Hasil yang diharapkan
08	Saat klik tombol Simpan akan menyimpan data baru yang sudah dibuat	Menyimpan data baru yang sudah dibuat
09	Saat klik tombol Kembali akan kembali ke tampilan atau menu awal	Bisa kembali ke tampilan atau menu awal
10	Saat klik Tombol Unggah akan memunculkan menu unggah dan mengunggah file yang sudah dipilih	Bisa menampilkan menu unggah dan mengunggah file yang sudah dipilih
11	Saat klik tombol Download akan mendownload file yang diinginkan	Bisa mendownload file yang diinginkan
12	Saat klik tombol Delete akan menghapus data yang diinginkan	Bisa menghapus data yang dipilih
13	Sengaja salah memasukan data	Keluar notifikasi tolakan yang berisi Data tidak valid
14	Sengaja salah mengupload data	Tombol unggah tetap disabled dan tidak enabled atau keluar notifikasi tolakan

From the table above describes the stages in conducting the test. testing the menu search by entering the code, "ASM4 code, the Registration menu will appear". testing by entering filter data and can appear, "Input component stock finder data then the desired component stock display will appear".

The next table from numbers 11, testing the buttons to go back, download, upload, and delete, "when you click the back button it will return to the main menu, when you click download it will automatically download the desired file, when you click upload it will upload the file which has been created in the correct format, when you click delete, the selected file will be deleted and cannot come back again (also applies to the database)".

Then test rejection if there are incorrect data. Checking the notification was successful and an error occurred. testing the results of the data entered into the database whether the same or not. testing of each existing menu such as the service menu example, the damage diagram master menu, the spare part price master menu, the visit area registration menu and the visit service product registration menu.

The flowchart in Figure 1 explains the sequence of the software testing work process at PT. Hartono Palace of Technology. Where the first thing to do is make a concept by the system. If it is finished, the concept will be reported in advance to the manager and a meeting will be held to discuss which part is not appropriate (if it is not appropriate, it will be revised continuously). If it has been accepted, then the next step is to give the draft file that has been made to the developer to work on later.
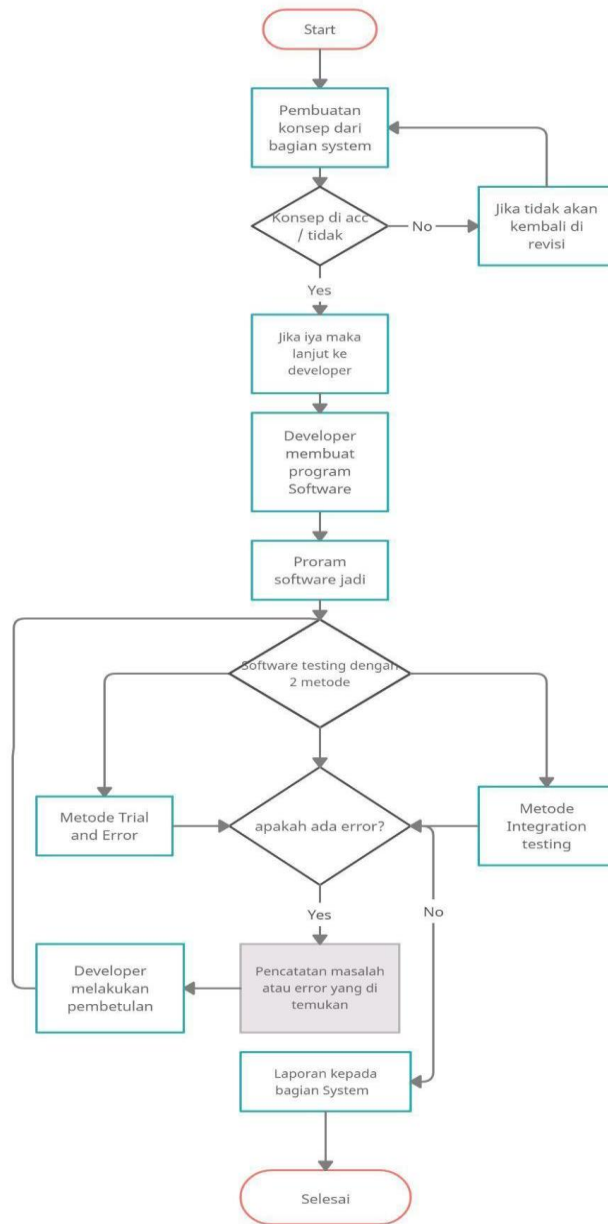

Figure 1. Software testing process Flowchart.

After being given to the developer, the developer will create a software program that is in accordance with the concept. Wait for the created software program to finish. If it is finished, the next step is the software testing process of the software program that was created earlier. In the testing process there are 2 methods used, namely Trial and Error and Integration testing. If a problem is found, there will be a record of the problem and it will be returned to the developer for repair. After that proceed to software testing again to check again. If no problems are found, a report is generated that the tested software program has no errors or problems and is ready to run.

IMPLEMENTATION AND RESULTS

IMPLEMENTATION
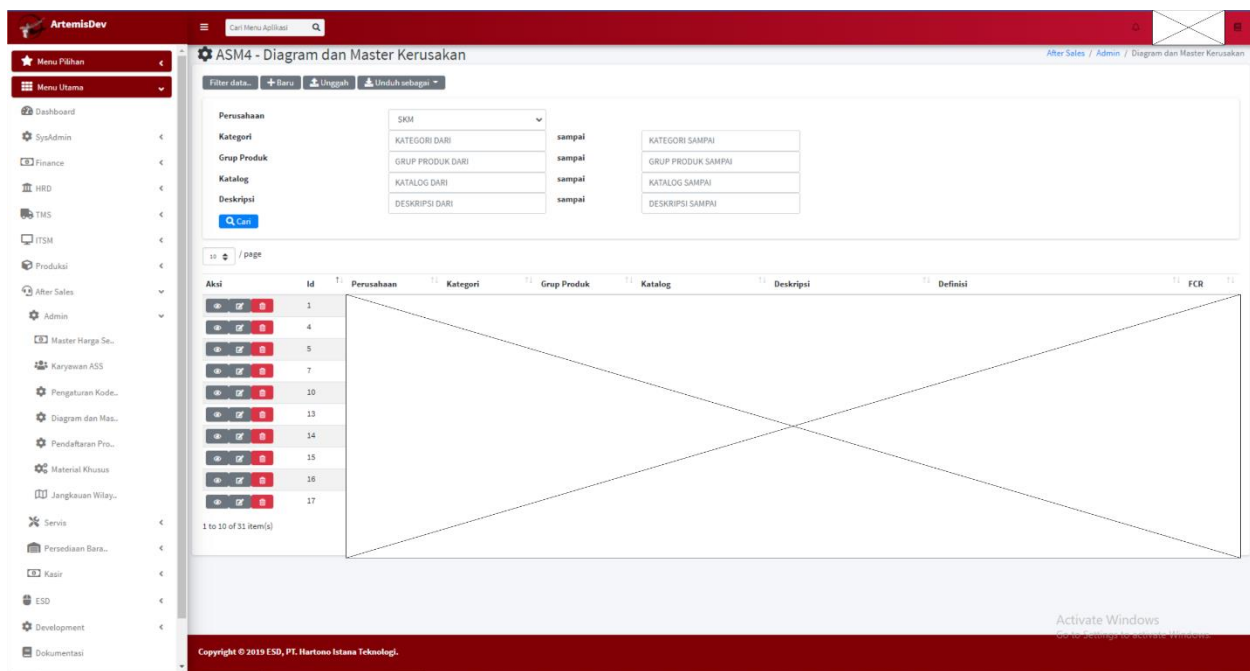
Figure 2.   Home Menu UI Display

Figure 2 describes an initial view of the menu that will be tested. To search for your own menu, you can use the existing search bar menu. For example, if we just enter the ASM4 code, the Diagram dan Master Kerusakan menu will appear. Then there is the Filter Data button which is useful for searching for existing data. On the Filter Data button menu there are sections that you want to find. For example there are Companies, Categories, Product Groups, Catalogs and Descriptions. Then at the bottom there are several buttons as well, namely Delete, Edit and buttons just to view. After the Filter Data button, there are several more buttons, namely New, Upload, and Download Buttons.

Next there is an overview of the menu display on the Create New button. In the new button menu display there are several things or sections that must be filled in according to existing needs.

The first to be filled are Company, Category, Product Group, Catalog, Description, Definition and the last FCR will be filled automatically if needed. In the Category and Product Groups section there is a dropdown. The dropdown itself will be tested whether it can display the data needed when creating new data.
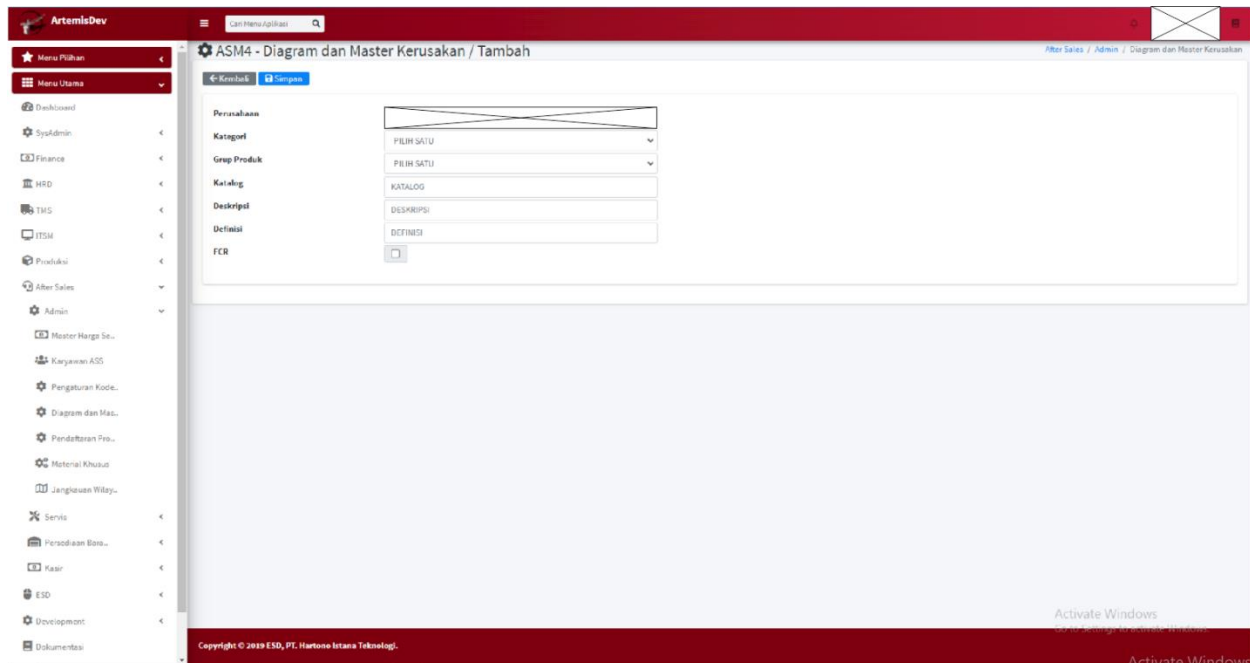

Figure 3.   New Button UI Appearance

Then the last sample image has an overview of the Upload button UI display menu. In the Upload menu, there are several buttons in it, namely the Template, Upload, and Save buttons. The template button itself is used to download files with excel format that have been provided to be filled and uploaded. Then there is the Upload button, the Upload button itself by default will be disabled at the beginning or access cannot. For the order of use itself, starting from the user downloading the template file that has been provided and filled in according to the data that has been determined. After that the user will upload the file that was created earlier by searching for the file in the File section and clicking on the Browse section to find the file to be uploaded. If the file to be uploaded has been selected, the Upload button will automatically be active and can be accessed to carry out the file upload process. After that, wait a few moments, the program will read whether the uploaded file matches the specified data. If the uploaded file is correct then the save button will be activated and can be accessed to save the correct file. However, if the wrong file is uploaded or the data contained in the file is still wrong, a rejection notification will appear that the uploaded file or data is still wrong and must be corrected before re-uploading.
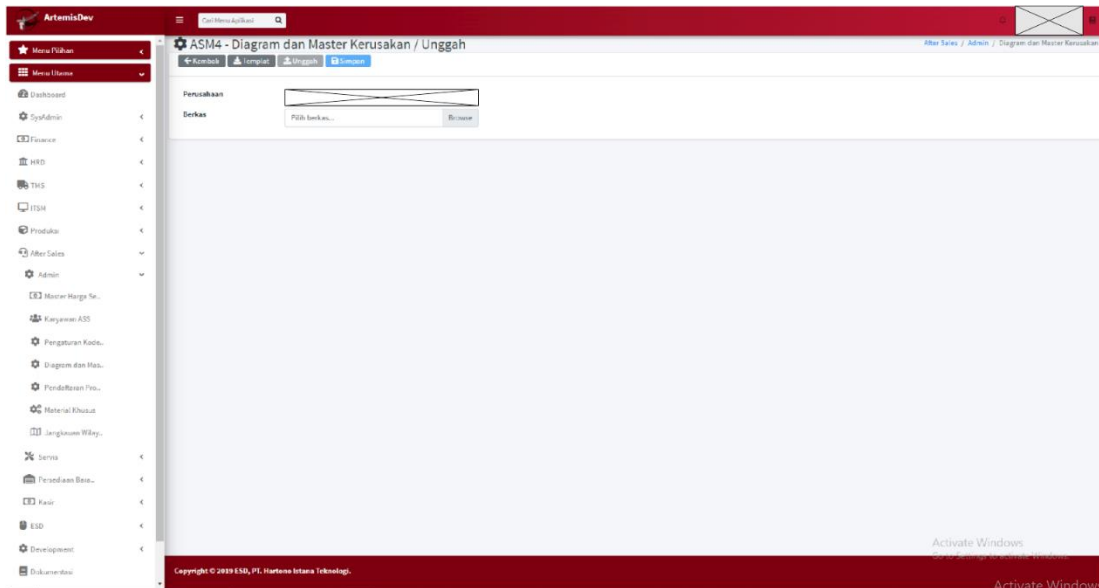
Figure 4. Upload button UI display

RESULTS

Figure 5 shows the test results from test cases that have been made previously. For the first, there is a bar chart of the Search Bar Menu which has been tested 10 times. The search bar menu itself is useful for searching for more specific menus, such as the Diagram menu and Damage Master by entering the specified code to search for the menu. For the number of menus sought there are 4 different menus. Each menu is tried 10 times and everything works so the diagram shows 100%.
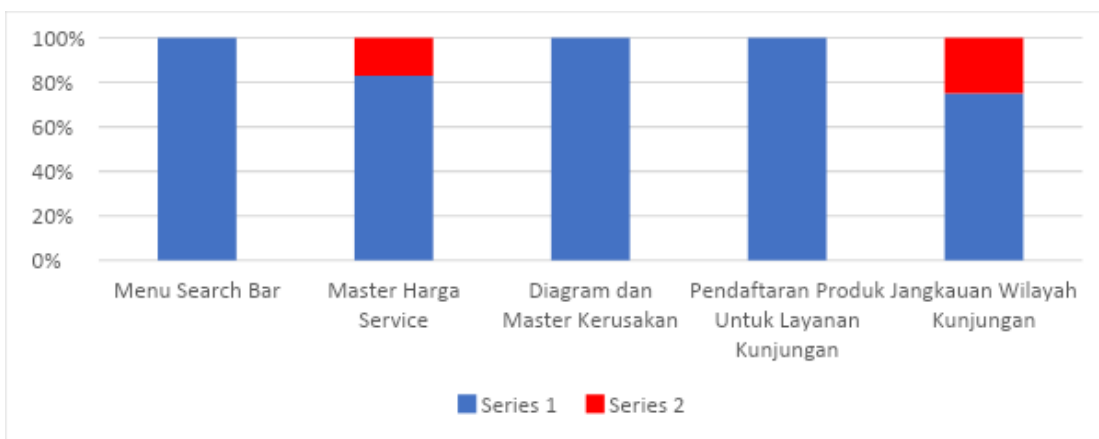
Figure 5. Table of test results for the Trial and Error Method

Next there is a bar chart from the Service Pricing Master menu. The test was also carried out 10 times. In testing this menu there are several buttons and data displays that must be tested. The first is the Filter Data button, it is used to search for data that is already in the menu and get 100% results. Next there is the Create New button. This button is useful for creating or adding new data

that is already set in it. Please note that each New Create button on some menus must be different because the data on each menu is different. Testing this Create New button got 100% result. Proceed to the next button, which is the Upload button. The Upload button itself is useful for uploading files with the excel format that has been provided in it and can be downloaded for templates. For test results, the Upload button itself gets 100% results. Furthermore, there is a Download button which is useful for downloading pre-existing data. For the test results the Download button itself gets 0% results or fails, because in the test there is an error in it so it can't download or download existing files. Then there is the Delete button which is useful for deleting the data you want to delete. On the Delete button itself there are 2 places that must be checked, namely on the initial menu display and the database. The Delete button alone gets 100% results. Finally on the Service Pricing Master menu there is a Database button. The Database button itself is useful for displaying in the Database section the data storage that has been created and deleted. The database button must be careful in testing because it relates to the correct data or not. For the Database button itself the result is 100%. So from all the results obtained from the Master Pricing Service menu with several buttons tested, the results were 83% successful and 17% failed.

Next there is the Diagram and Master Damage menu. The test was also carried out 10 times. In testing this menu there are several buttons and data displays that must be tested. The first is the Filter Data button, it is used to search for data that is already in the menu and get 100% results. Next there is the Create New button. This button is useful for creating or adding new data that is already set in it. Testing this Create New button got 100% result. Proceed to the next button, which is the Upload button. The Upload button itself is useful for uploading files with the excel format that has been provided in it and can be downloaded for templates. For the test results, the Upload button itself gets 100% results. Furthermore, there is a Download button which is useful for downloading pre-existing data. For the test results the Download button itself gets 100% results. Then there is the Delete button which is useful for deleting the data you want to delete. On the Delete button itself there are 2 places that must be checked, namely on the initial menu display and the database. The Delete button alone gets 100% results. Finally on the Service Pricing Master menu is the Database button. The Database button itself is useful for displaying in the Database section the data storage that has been created and deleted. Database button must be careful in testing because it deals with correct data or not. For the Database button itself the result is 100%. So from all the results obtained from the Master Price Service menu, with several buttons tested the results were 100% successful and 0% failed.

Next there is a Product Registration menu for Visiting Services. The test was also carried out 10 times. In testing this menu there are several buttons and data displays that must be tested. The first is the Filter Data button, it is used to search for data that is already in the menu and get 100% results. Next there is the Create New button. This button is useful for creating or adding new data that is already set in it. Testing this Create New button got 100% result. Proceed to the next button, which is the Upload button. The Upload button itself is useful for uploading files with the excel format that has been provided in it and can be downloaded for templates. For test results, the Upload button itself gets 100% results. Furthermore, there is a Download button which is useful

for downloading pre-existing data. For the test results the Download button itself gets 100% results. Then there is the Delete button which is useful for deleting the data you want to delete. On the Delete button itself there are 2 places that must be checked, namely on the initial menu display and the database. The Delete button alone gets 100% results. Finally on the Service Pricing Master menu there is a Database button. The Database button itself is useful for displaying in the Database section the data storage that has been created and deleted. The database button must be careful in testing because it relates to the correct data or not. For the Database button itself the result is 100%. So from all the results obtained from the Master Pricing Service menu, with several buttons tested the results are 100% successful and 0% failed.

Finally, there is the Range of Visiting Areas menu. The test was also carried out 10 times. In testing this menu there are several buttons and data displays that must be tested. The first is the Filter Data button, it is used to search for data that is already in the menu and get 100% results. Next there is the Create New button. This button is useful for creating or adding new data that is already set in it. Testing the New Create button gets 0% results or fails because when displaying the required data it does not appear. Proceed to the next button, which is the Upload button. The Upload button itself is useful for uploading files with the excel format that has been provided in it and can be downloaded for templates. For the test results, the Upload button itself gets 50% results, this upload button should work but there is a format error in the excel template used so that the file is correct but when uploaded is still wrong. Furthermore, there is a Download button which is useful for downloading pre-existing data. For the test results the Download button itself gets 100% results. Then there is the Delete button which is useful for deleting the data you want to delete. On the Delete button itself there are 2 places that must be checked, namely on the initial menu display and the database. The Delete button alone gets 100% results. Finally on the Service Pricing Master menu is the Database button. The Database button itself is useful for displaying in the Database section the data storage that has been created and deleted. Database button must be careful in testing because it deals with correct data or not. For the Database button itself the result is 100%. So from all the results obtained from the Master Price Service menu with several buttons tested, the results were 75% successful and 25% failed.

TEST TRUTH BENCHMARK

This section discusses the standard of truth that was carried out for this test. The goal is that the reader does not wonder about the standard of truth of this test. The following is an example of each menu.

Menu Master Kerusakan

Figure 6. Benchmark of truth for search bar menu testing

In the concept for the Menu Search Bar section, all menu concept models must be the same, so here I take the Menu Search Bar section. At the beginning, there must be an overview of the concept of Access Role, the purpose of which is to give access rights to certain people so that they can enter certain menus by setting the access role. Then there is the User Role, which is useful for knowing the type of rights and the type of user from the person who wants to access the menu. Then there is a code to access a certain menu (each menu has a different code).

Figure 7. Result benchmarks for the correctness of the menu master kerusakan

The picture above is the result of testing on the damage master menu. Because when it was checked, it was in accordance with the standard of truth in the concept design. So what is written is only notes and input when the test is finished.

Menu Master Harga Sparepart dan Jasa

Figure 8.   Benchmark of truth for testing Data Filter

For each concept the Data Filter on each menu must be the same. shown in Figure 8. i.e there is an option to display some data in another menu and match whether it is the same as that in the menu other than this data filter. Then there is information on the side to make it easier during the testing process by looking at the benchmark for the truth of the information available.

Figure 9.   Result benchmarks for the correctness of the menu master harga spare part dan jasa

The picture above is the result of testing on the Master Menu for Spare Parts and Services Prices. Actually, there are some buttons as well which is tested but in this case, it is taken on the create new button. Initially, there was an error in the test after that it was recorded in the report.

After that, it has just been corrected by the developer, it will automatically be replaced in the report "already corrected".

Menu Pendaftaran Jangkauan Wilayah Kunjungan

Benchmark for the correctness of the test on the registration menu for the coverage area of the visit.
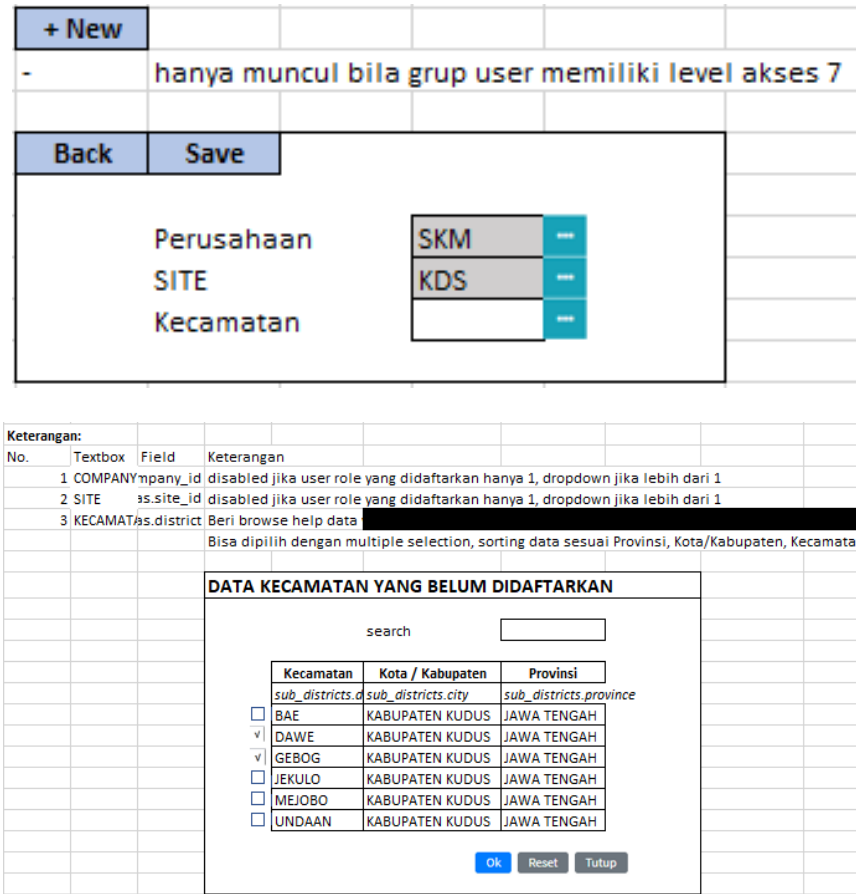

Figure 10.   Benchmark the truth of the +New button test

For the +New button concept itself, initially, it will be ensured that the user who will access it must have a certain level. The level setting is at the beginning, namely on the Access Role menu. Furthermore, the button that will display the data that will be used and the data taken or integrated from another menu must be checked whether the data displayed on the button is the same or not. Next, there is an explanation of each part or button. Then there are examples of the results that have been created.
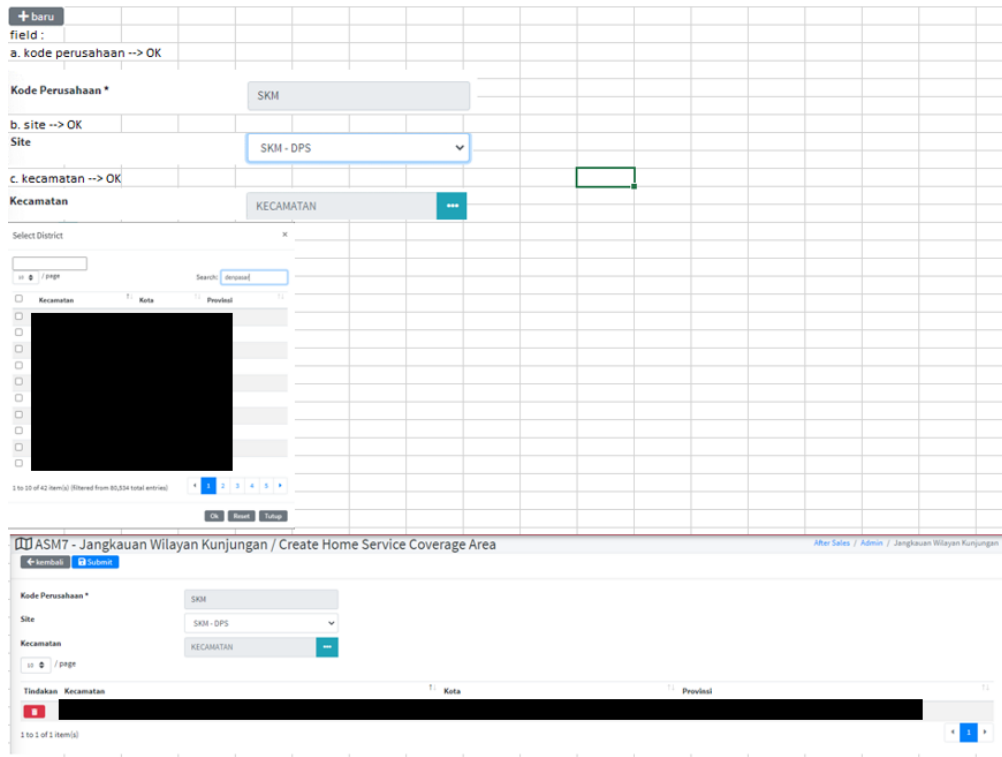
Figure 11. Result Benchmarks for the correctness of the Menu Pendaftaran Jangkauan Wilayah Kunjungan

Figure 11 is the result of testing on the Registration Menu for Visiting Areas. Actually, there are some buttons as well which is tested but in this case, it is taken on the create new button. Because there were no errors during the test, the examiner gave an "OK" statement.

Menu Pendaftaran Produk Untuk Layanan Kunjungan

Lastly, there is the concept of an Upload button. The first is an overview of the Upload menu which consists of several buttons. Then there is an explanation of the function of each button and there is also a rejection notification if an error occurs during upload. This chapter describes the standard of correctness of each menu tested. Each menu that is tested, must have almost the same concept picture. So here I only take a few examples of benchmarks. The tests carried out are based on concepts that already exist or have been created. So the results of the existing diagram table are the results of testing based on existing concepts and tested with two different methods.
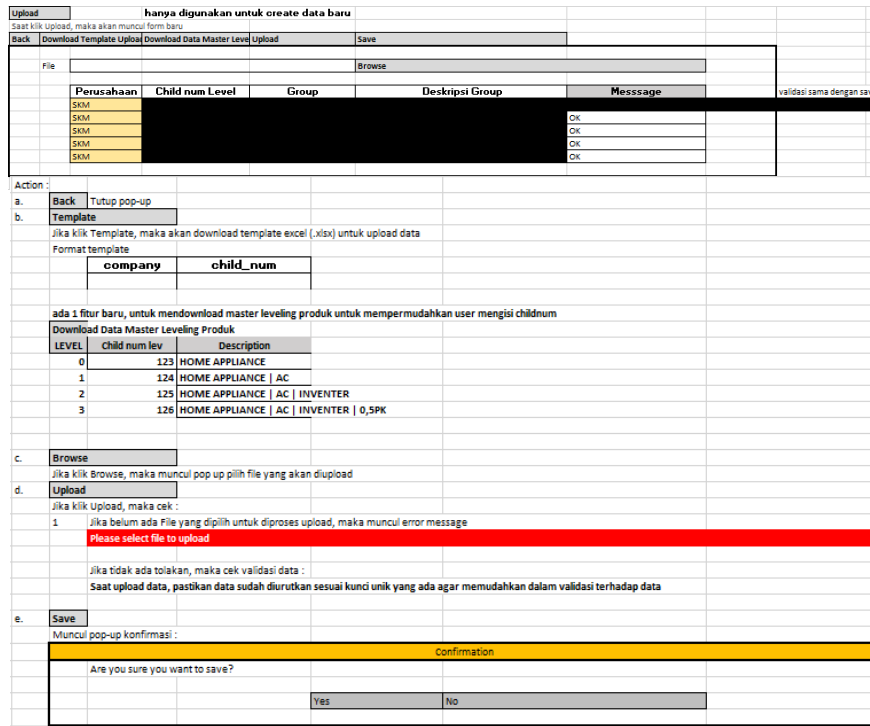
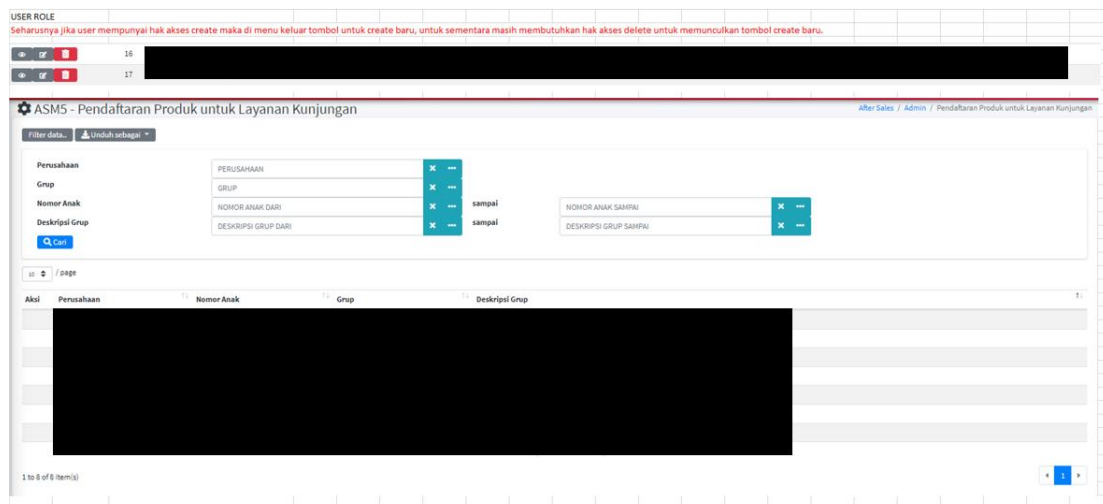**Figure 12.** Benchmark of truth for Upload button test

Figure 13. Result benchmarks for the correctness of Menu Pendaftaran Produk Untuk Layanan Kunjungan

Figure 13 is the result of testing on the Product Registration Menu for Visiting Services. Actually, there are several buttons that have been tested, but in this case, the access rights and data filter buttons are being tested. In testing, an error was found in setting the access rights of the user role, so in the report, it is explained where the error is and is marked in red.

BINARY CLASSIFICATION

Trial and error method

Tabel 2. Confusion Matrix Trial and Error method

		Predicted	
		Positive	*Negative*
Actual	*Positive*	220	0
	Negative	30	0

Table 2, shows the confusion matrix for trial and error method. From this matrix we can calculate the overall performance according to the trial and error method as shown in Table 3.

Tabel 3. Performance using Trial and Error Method

Measurement	Score
Precission	0.88
Recall	1
Accuracy	0.88

Integration Testing method

Tabel 4. Confusion Matrix Trial and Error method

		Predicted	
		Positive	*Negative*
Actual	*Positive*	130	0
	Negative	30	0

Table 4, shows the confusion matrix for trial and error method. From this matrix we can calculate the overall performance according to the trial and error method as shown in Table 5.

Tabel 5. Performance using Trial and Error Method

Measurement	Score
Precission	0.75
Recall	1
Accuracy	0.75

In the Trial and Error method, the recall calculation is 100% and the precision is 88%. The results obtained are 220 True positive and 30 False positive, while for False negative and True negative the number is 0. Taken from testing the integration testing method from 5 different menus.

In the integration test method, the recall calculation is 100% and the precision is 81%. The results obtained are 130 True positive and 30 False positive, while for False negative and True negative the number is 0. Taken from testing the integration testing method from 4 different menus. True Positive itself is when the prediction is successful and the actual value is also successful. For False Positive itself is when the prediction is successful but for real results it fails or does not work. The results obtained from this test are included in the High Recall High Precision category. High Recall itself prefers False Positive to be better than False Negative, therefore in this test there is no False Negative found. For High Precision itself, it is more directed to the occurrence of True Positive so that it does not want False Positive. This test is more towards Precision because of the higher number of True Positives.

CONCLUSION

At the end of this report, the author will describe some conclusions that can be drawn and the suggestions contained in the research results. Broadly speaking, the author concludes that the test results from several menus including Master Harga Service, Diagram dan Master Kerusakan, Pendaftaran Produk Untuk Layanan Kunjungan, dan Jangkauan Wilayah Kunjungan with two methods have obtained the same results. More specifically, the author can draw the following conclusions:

1. Conducted testing on several menus including Master Harga Service, Diagram dan Master Kerusakan, Pendaftaran Produk Untuk Layanan Kunjungan, dan Jangkauan Wilayah Kunjungan with two different methods but with the same result. this can be seen from several aspects including the Trial and Error method which is a thorough testing stage without leading to one thing and the Integration testing method which is more inclined to testing the relationship between the menu and the resulting data. So that the Trial and Error method itself is easier in software testing.

2. In terms of choosing which method is the most appropriate. Of course both methods are correct. But it depends on the context you want to test. In this case, it has been explained that the Trial and Error method is more comprehensive in software testing while for integration testing itself it is more towards the relationship between menus and data display. So that in terms of software testing at PT Hartono Istana Teknologi, the Trial and Error method is sufficient for software testing. This meaning can be seen from how successfully found or not an error in a software. From testing the Trial and Error method, more test results were obtained. From testing the Integration test method itself, getting less test results. Because this method tends to lead to the display of correct data results or not. While the Trial and Error method is more directed at the concepts that have been made.

REFERENCES

[1] M. Divo, "Optimasi Pencampuran Batubara Beda Kualitas Dengan Metode Trial And Error untuk Memenuhi Kriteria Permintaan Konsumen di CV. Bara Mitra Kencana Kota Sawahlunto Sumatera Barat," vol. 5, no. 1, p. 10.

[2] P. Duval and M. Lomperski, "Control System Evolution and the Importance of Trial and Error," *Control Systems*, p. 4, 2017.

[3] K. Sekarsari and T. Tata, "Performance analysis of PID control in DC Brushless motor using trial and error method," *IOP Conf. Ser.: Mater. Sci. Eng.*, vol. 1098, no. 4, p. 042027, Mar. 2021, doi: 10.1088/1757-899X/1098/4/042027.

[4] T. Imai, H. Masuhara, and T. Aotani, "Making live programming practical by bridging the gap between trial-and-error development and unit testing," in *Companion Proceedings of the 2015 ACM SIGPLAN International Conference on Systems, Programming, Languages and Applications: Software for Humanity*, Pittsburgh PA USA, Oct. 2015, pp. 11–12. doi: 10.1145/2814189.2814193.

[5] F. Ipate and M. Holcombe, "An integration testing method that is proved to find all faults," *International Journal of Computer Mathematics*, vol. 63, no. 3–4, pp. 159–178, Jan. 1997, doi: 10.1080/00207169708804559.

[6] M. Jaffar-ur Rehman, F. Jabeen, A. Bertolino, and A. Polini, "Testing software components for integration: a survey of issues and techniques," *Softw. Test. Verif. Reliab.*, vol. 17, no. 2, pp. 95–133, Jun. 2007, doi: 10.1002/stvr.357.

[7] U. Linnenkugel and M. Mullerburg, "Test data selection criteria for (software) integration testing," in *Systems Integration '90. Proceedings of the First International Conference on Systems Integration*, Morristown, NJ, USA, 1990, pp. 709–717. doi: 10.1109/ICSI.1990.138737.

[8] W. T. Tsai, Xiaoying Bai, R. Paul, Weiguang Shao, and V. Agarwal, "End-to-end integration testing design," in *25th Annual International Computer Software and Applications Conference. COMPSAC 2001*, Chicago, IL, USA, 2001, pp. 166–171. doi: 10.1109/CMPSAC.2001.960613.

[9] S. P. Shashank, P. Chakka, and D. V. Kumar, "A systematic literature survey of integration testing in component-based software engineering," in *2010 International Conference on Computer and Communication Technology (ICCCT)*, Allahabad, Uttar Pradesh, India, Sep. 2010, pp. 562–568. doi: 10.1109/ICCCT.2010.5640467.

[10] Youngchul Kim and C. R. Carlson, "Scenario based integration testing for object-oriented software development," in *Proceedings Eighth Asian Test Symposium (ATS'99)*, Shanghai, China, 1999, pp. 283–288. doi: 10.1109/ATS.1999.810764.