Jurnal Teknik Informatika (JUTIF)
Vol. 5, No. 3, June 2024, pp. 757-765

DOI: https://doi.org/10.52436/1.jutif.2024.5.3.1848
p-ISSN: 2723-3863
e-ISSN: 2723-3871

REUSE OF THE EFSM MODEL OF PEDULILINDUNGI APPLICATION IN SATUSEHAT APPLICATION TESTING WITH MBT METHOD

Muamar Fajar Rahmadani[*1], Rosa Reska Riskiana[2], Dana Sulistyo Kusumo[3]

[1,2,3]Informatics School of Computing, Telkom University, Indonesia
Email: [1]muamarfajar@student.telkomuniversity.ac.id, [2]rosareskia@telkomuniversity.ac.id, [3]danakusumo@telkomuniversity.ac.id

(Article received: January 18, 2024; Revision: February 6, 2024; published: May 28, 2024)

Abstract

On 2023, the Government of Indonesia announced the change of PeduliLindungi application to SatuSehat, with the addition of features that have been integrated with Electronic Medical Records (RME). In this research, the concept of model reuse is applied to facilitate the creation of test models on the same features between PeduliLindungi and SatuSehat, namely Linked Profile and Covid-19 Vaccine. In applying the reuse model, the method template and edge template strategies are used to adjust to the evolution of the model that occurs in the SATUSEHAT application, in the edge template or second iteration there are additional vertices and edges on the Linked Profile and Vaccine features. By combining the number of vertices and edges, the overall similarity percentage is around 79.81% on the Linked Profile feature, showing the efficiency of modeling with a reuse model of around 20.19%. Testing on SatuSehat using Altwalker tools with Random and Weighted Random algorithms shows high coverage achievements, especially on vertex, these achievements show the effectiveness of the reuse model. Comparison with previous research on PeduliLindungi shows an increase in coverage rate, especially on features that apply the reuse model. This research illustrates the success of the reuse model concept in accelerating the development of test models and increasing coverage in applications where changes occur.

Keywords: *Altwalker, Extended Finite State Machine (EFSM), Model Based Testing (MBT), Model Reuse, SATUSEHAT.*

1. INTRODUCTION

In early 2023, the Government of Indonesia announced the change of PeduliLindungi application to SatuSehat. The main purpose of the change is to expand the functionality of the app by adding features that integrate with Electronic Medical Records (RME) in hospitals [1]. The evolution from PeduliLindungi to SATUSEHAT reflects the government's efforts to improve digital health services and provide easy access to health information to the public.

The PeduliLindungi application was first released in 2020, which has main features that have features that perform operations or changes to Create, Read, Update, and Delete (CRUD) data such as Linked Profiles, covid-19 vaccines, and Ehac. In early 2023, SatuSehat was presented as a replacement for PeduliLindungi with more complete features, because it has been integrated with Electronic Medical Records (RME) as well as health services and Reminders to Take Medication [2]. In addition to these features, there are features that are still maintained from PeduliLindungi, namely the Covid-19 Vaccine and Linked Profile, but the Linked Profile in SATUSEHAT has changed its name to Linked Profile. In the similarity of the features of these two applications, especially in the vaccine and Linked Profile features, there is an additional flow to these features. Based on the similarity of features between PeduliLindungi and SatuSehat, the author carries out the use of the reuse model in this study, because the PeduliLindungi application has been tested with Model Based Testing in Maulidito Dwinandana's research, so the question arises regarding the use of existing models in Maulidito Dwinandana's research can be reused in SATUSEHAT testing, especially in the Linked Profile and Covid-19 Vaccine features [3].

Model reuse is a concept that involves reusing test models that have been designed using method template and edge template strategies [4]. The method template will use the previous test model as the first iteration, and then the edge template is used to generate the evolution of the test model in the second iteration. This concept can facilitate the development of test models for SatuSehat because model creation is based on the template model that has been made in Maulidito Dwinandana's research [3], and applying the second iteration will add new edges as an adjustment to the SATUSEHAT application, so that using this reuse model concept can speed up and simplify the testing process [5].

The testing method that will be used in the SatuSehat application is automated testing. In this research, the testing approach used is Black-Box Testing, where in black box testing there are several

techniques, one of which is Model-Based Testing (MBT) [6]. MBT is the choice in this research as a research method because, MBT is a testing method that uses a state machine model to build fast test cases [7]. This research took inspiration from the journal Extended Finite State Machine on PeduliLindungi written by Maulidito Dwinandana. In the research, it has been proven that MBT is effective in overcoming the CRUD feature in the application optimally and providing fast execution time by using EFSM (Extended Finite State Machine) modeling in the PeduliLindungi application. This research will use Altwalker tools as a tool for MBT execution, where Altwalker has an executor algorithm that implements Graphwalker, so that it can get optimal test results [8], [9].

Model-Based Testing uses several methods which are distinguished in designing test models, models can be represented in several methods such as Finite State Machine (FSM), Extended Finite State Machine (EFSM), and Markov Chain [10]. Testing in the SATUSEHAT application uses MBT with the Extended Finite State Machine method which is a Finite State Machine method that uses a data operation at the model transition [11]. Since the following research is based on the reuse model in PeduliLindungi's research written by Maulidito Dwinandana, testing will also use the Random and Weighted_Random algorithms which can achieve the testing goal of achieving a high percentage of coverage [12]. By using the EFSM method, the Random and Weighted_Random algorithms will be clearly visible regarding the running scenario of the model because EFSM also affects the input and output of each transition between models [13].

In this research applying the concept of reuse model on SatuSehat application testing using MBT - EFSM as automated testing, with Altwalker tools. This research seeks to accelerate the development of a testing model for SatuSehat with the concept of a reuse model, because the template model from previous research will speed up testing because the test cases that will be generated can be reused or the test cases produced will not be much different [14]. This research will evaluate the results of applying the reuse model to the SATUSEHAT application model by covering high coverage after model evolution with the reuse model concept by using Random and Weighted_Random algorithm based on how it is effective for coverage percentages [15].

This research topic focuses on the implementation of Model Reuse with MBT-Extended Finite State Machine (EFSM) in the context of SATUSEHAT application, with the aim of implementing model reuse by using existing method templates (model templates) in previous research. The main questions raised were about the possibility of implementing this model in the SATUSEHAT application and the process of implementing the reuse model in this study. Problems encountered include the registration feature not being part of the test as the registration system is already integrated with the NIK, and only certain features such as Linked Profile, Covid-19 Vaccine, and the new Reminder to Take Medication feature will be tested, in accordance with the features that previously existed in the PeduliLindungi application.

This research approach explores the application of Model Reuse with MBT-EFSM as a solution to reduce testing time on the SATUSEHAT application. By limiting the scope to key features relevant to the previous application and considering certain technical constraints, this research aims to contribute to the development of efficient and reliable testing methods, particularly in the Android environment. Through this research, it is expected that a solution will be found that can serve as a foundation for improving the quality of testing on the SATUSEHAT application.

This research aims to find out the Model Reuse technique can be implemented in Model-Based Testing with Extended Finite State Machine, and compare the results of this test against previous research conducted by Maulidito Dwinandana.

2. RELATED WORKS

2.1. Model Based Testing (MBT)

Model-Based Testing (MBT) is a technique that uses system models to generate and execute test cases automatically. MBT is a branch under black-box testing. The use of MBT is motivated by observations from manual testing, where manual testing is unstructured, undocumented and dependent on testers [16]. In general, MBT steps begin with building a System Under Test (SUT) based model for testing, then selecting test criteria to be used to develop test case specifications. Then the selected model and test specifications are used to generate test cases. Finally, the test cases are executed and the results of the execution are validated. A detailed description of MBT can be found in [10].

In Research [3], explained that there are main differences between Model-Based Testing and Black-box Testing, namely in the creation of tests, where Black-box is still done manually and Model-Based Testing makes tests from the model automatically [3] The model created by MBT is a state machine, flowchart, or UML diagram, which is obtained based on system requirements. Test cases are automatically created by adding several criteria, ranging from constraints, input/output, and coverage.

2.2. Extended Finite State Machine (EFSM)

Research [17] states that the EFSM is an extension of the traditional FSM to overcome the shortcomings of the traditional FSM. This is achieved by adding trigger conditions that need to be met in order for a transition between states to occur. EFSM extends FSM by modeling a system with control and

data parts (unlike FSM which can only model the control part of a system). It used to be adopted to solve the problem of manually generating test cases [18].

The Extended Finite State Machine method is similar to the Finite State Machine, where the difference between the two methods is in the transition, which has an action syntax, where the parameter states an if condition that determines the condition to perform the transition if the if statement is true, and the action syntax is the output produced when a condition is true or false. Figure 2.3.1 and Figure 2.3.2 will explain and compare how the EFSM model and FSM model work.

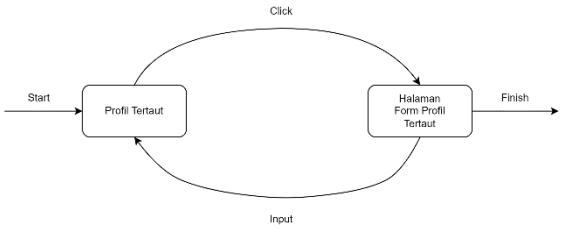

Figure 2.3.1 Finite State Machine

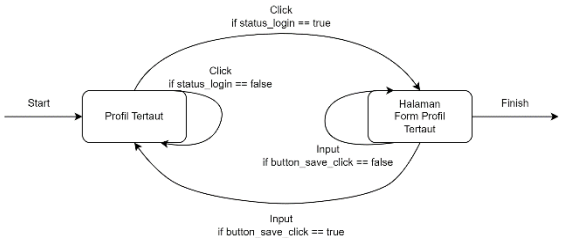

Figure 2.3.2 Extended Finite State Machine

Figure 2.3.1 shows an example of a Finite State Machine model on the Linked Profile input system. In the initial state the user will be directed to the Linked Profile page and when there is a Click event then the state will move from the Linked Profile to the Linked Profile Form Page. The Linked Profile Form Page will move to the Linked Profile state if the Input event is called. While the EFSM in Figure 2.3.2 shows the use of parameters in the model where if the parameters of the transition are met then the previous event will move to the next event based on the parameters of each transition and if the parameters are not met then the event will not move or fail [9], [13]..

2.3. Altwalker

Altwalker, a test execution tool developed by Altom, utilizes model-based testing using GraphWalker. It performs test path execution from a directed graph finite state machine, where each model is mapped into a class with the same name. The edges and vertices in the graph are also mapped as methods in the corresponding classes. The inspiration for the test structure comes from Python's unittest module approach. By using an Extended Finite State Machine (EFSM) to define the model, AltWalker allows the use of variables and guards to control transitions between states. AltWalker also provides tools to create, visualize, and validate models online or in Visual Studio Code [19].

2.4. Model Reuse

Model reuse in model-based testing is a concept to reuse models that have been created for previous systems or products in the testing process. Model reuse can save testing time, and improve test quality and consistency. Model reuse can be done in various ways, such as one of them using the method template and edge template strategies, where the method template as a strategy for taking the structure of the previous test model, then the edge template as a strategy for developing the model to adjust to the needs of the application. Method templates outline the process of selecting inputs, determining output expectations, and creating test cases based on the model structure, while edge templates navigate how the model runs based on the states and transitions that are passed so that a scenario can be created to verify the expectations of the model results [5]. Model reuse also requires techniques and tools that can support the creation, selection, configuration, and validation of models effectively and efficiently. Model reuse is one of the challenges and opportunities in the development of model-based testing [5].

2.5. Appium

Appium is an open source automation testing tool for native, mobile web, and hybrid apps on iOS and Android. Appium uses a Client-Server Architecture. Automation testing works with the REST API. There are many client libraries in Java, Ruby, Python, PHP, JavaScript, and C# that support Appium extensions. Appium Server is written in Node.js. The server can be installed directly using NPM. The server accepts connections from clients, and then translates and executes commands to provide responses as a result of command execution [20].

2.6. SATUSEHAT

SATUSEHAT Mobile is a public health service application previously named PeduliLindungi, which during the Covid-19 pandemic became a means to facilitate the delivery of information related to the development and control of Covid-19 to the Indonesian people. In March 2023, the Government of Indonesia, specifically the Ministry of Health of the Republic of Indonesia officially updated the PeduliLindungi application to become the SATUSEHAT health application, the update made to SATUSEHAT is an integration with data at the hospital, namely a new feature that has integrated individual health data between health service facilities (fasyankes) in the form of electronic medical records (RME) with various features and health services whose data is sourced and integrated with the

SATUSEHAT Platform, so that these changes make it easier for the public to access medical record data from hospitals. Features that previously existed in SATUSEHAT, such as Chek in, Vaccine Information, Covid-19 Test Results, and Health Services [1].

3. SYSTEM BUILT

There are several stages in conducting this final project research process. The following is the process of the system stages built in the author's final project research, based on the STLC stages:

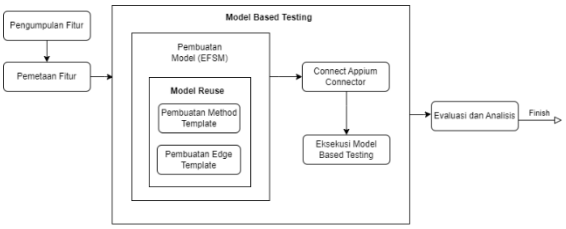

Figure 3. Process of the System Stages.

3.1. Feature Identification

At this stage, the main features of the SATUSEHAT application are identified, which include the CRUD system and which features are similar to PeduliLindungi. This feature selection is based on the features contained in Maulidito Dwinandana's research [2], but there is one feature that has been changed, namely the Ehac feature to Reminder to Take Medicine, because in SATUSEHAT this feature is no longer used in flights in Indonesia so the feature is removed, and features that get negative feedback on the SATUSEHAT application from the play store. The following are the features in Maulidito Dwianandana's research and the features that received negative feedback from the Play Store.

Table 3.1.1 PeduliLindungi CRUD Features

No	Feature	Description
1	Linked Profile	Create and Delete Profile
2	Vaksin Covid-19	Create and delete Overseas Travel Certificate
3	Ehac	Create and delete eHAC by vehicle

Table 3.1.2 SATUSEHAT CRUD Features

No	Feature	Description
1	Profil Tertaut	Add and remove Profil Tertaut
2	Vaksin Covid-19	Download Vaksin
3	Pengingat Minum Obat	Create and Delete data Pengingat minum obat

In table 3.1.1 are the features tested in Maulidito Dwinandana's research, of the three features, the Linked Profile feature and the covid-19 vaccine are still used in his latest application, SATUSEHAT. The Ehac feature previously found in PeduliLindungi was useful for monitoring the travel of Indonesian citizens to reduce the spread of the virus, when it changed to SATUSEHAT, the feature was removed because Ehac was no longer a requirement for flights in Indonesia.

Table 3.1.2 shows the features contained in SATUSEHAT, where the features are still used from PeduliLindungi, the features that are still used, namely the previously Linked Profile and covid vaccine - 19. Both features also have the same flow but there are some additional flows due to the evolution of these features, so adjustments need to be made. The Reminder to Take Medicine feature is a replacement for the Ehac feature because the feature has been removed to adjust to Indonesian aviation regulations. After identifying the features to be tested, table 3.1.2 is the features that will be tested in this study.

3.2. Feature Mapping

At this stage, the results of the identification of features that have previously been carried out will then be mapped into the vertex and edge of the features in table 3.1.2. Where vertex naming begins with the letter "v" and is followed by the application status, for edge naming begins with the letter "e" and is followed by verbs, namely click and back. The following are the features that have been mapped based on the feature table 3.1.2.

Table 3.2 Feature Mapping

No	Vertex and Edge	Label
1	Profil Tertaut Page	v_ ProfilTertaut
2	Vaksin Page	v_Vaksin
3	Pengingat Minum Obat Page	v_PengingatMinumObat
4	Click Profil Tertaut	e_ClickProfilTertaut
5	Click Vaksin	e_ClickVaksin
6	Click Pengingat Minum Obat	e_ClickPengingatMinumObat

Table 3.2 is the result of mapping based on feature table 3.1.2, namely adding the letter "v" for the vertex prefix to the feature name, as seen in the drug reminder feature "v_ReminderMinumObat" and the letter "e" for the edge prefix, such as "e_ClickReminderMinumObat" with this mapping makes it easier to create an EFSM model because it has been distinguished between vertex and edge.

3.3. EFSM Model Creation

The model creation this study begins by creating a model using a flowchart that will represent how the SATUSEHAT application runs from the beginning to the features that have been mapped in table 3.2, based on the user's point of view and will be combined with the 3 features that will be tested in one flowchart. Figure 3.3.1 is the flow of modeling based on the SATUSEHAT application using a flowchart.

The flow of the SATUSEHAT model creation is from user perspective where models are created based on how user use the SATUSEHAT application. Based on that, the model does not use SRS or SDD of SATUSEHAT application, the model shows that the state are pages and the transition are the action that

the user will encounter on every pages. User can access or return from almost every page with their action names which some of them required a condition, so the model will need to be created with a condition on the edge by using EFSM for every action.

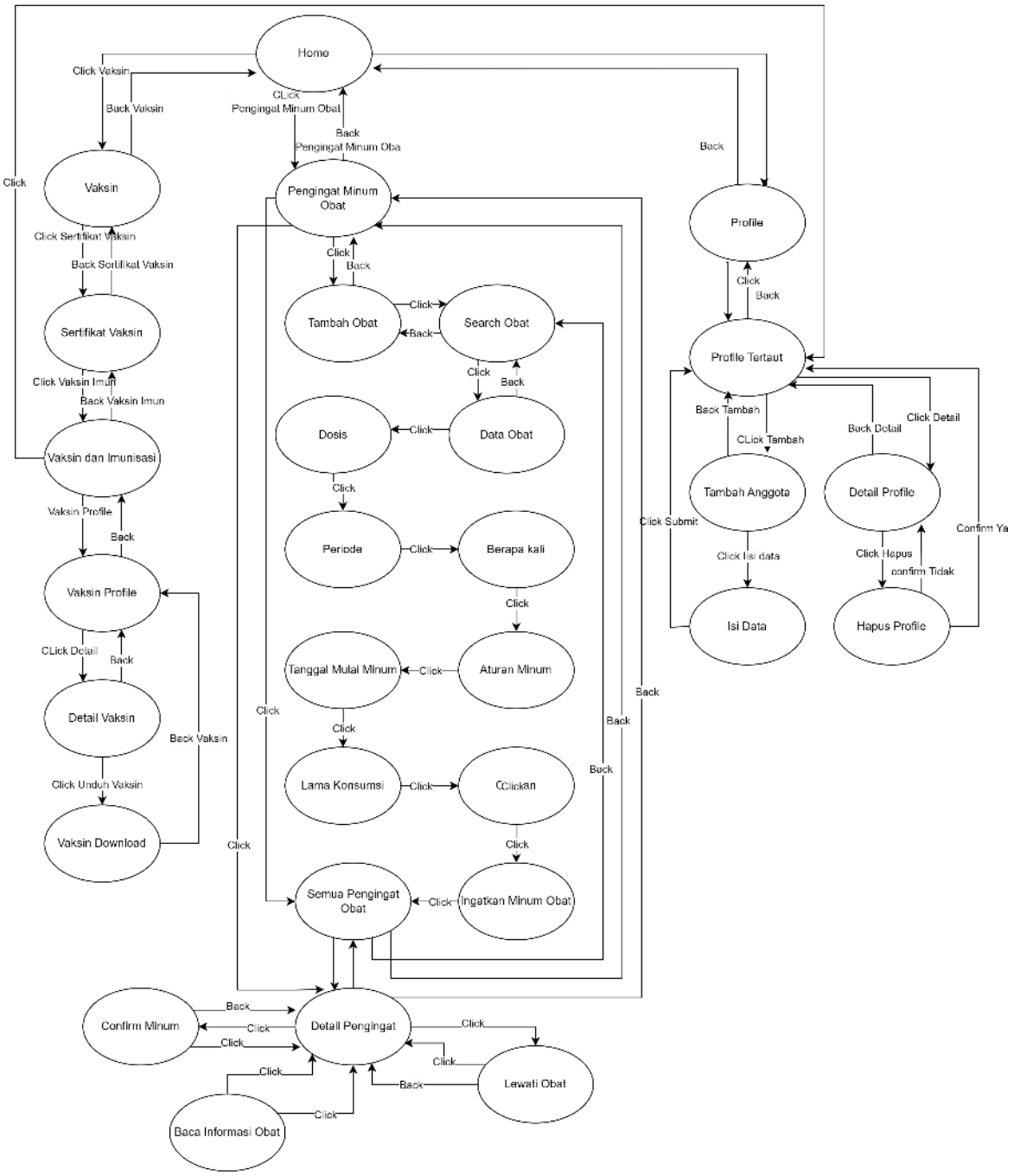

Figure 3.3.1 Flow of the SATUSEHAT model creation

In making the EFSM model using the Altwalker editor model by applying the reuse model. With the reuse model approach, it can reduce time during model development, because modeling is based on models that have been tested in Maulidito Dwinandana's research, which will then be modified to suit the needs of the SATUSEHAT application. In model creation, naming is divided into two, namely "v" and "e". Where "v" represents a vertex that determines the state that determines the current state and "e" as an edge that represents an event in a transitions. Modeling is done based on how the SATUSEHAT application runs where there are 3 main features for application testing which are marked using State and represented on the application page. The following is an example of an EFSM model of the features that have been mapped in Table 3.2. Figure 3.3.2 is an example of feature modeling in SATUSEHAT after the feature mapping stage.

DOI: https://doi.org/10.52436/1.jutif.2024.5.3.1848
p-ISSN: 2723-3863
e-ISSN: 2723-3871

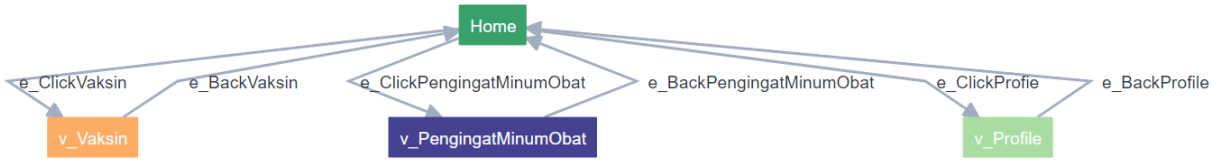

Figure 3.3.2 Example of Feature Modeling on SatuSehat

In Figure 3.3.2 is an example of modeling with the EFSM approach on the home page which will lead to the 3 main features that have been selected. The naming in the model is divided into two, namely v and e, v as a vertex that determines the current state and e as an edge that labels events in a transition. Seen in Figure 3.3.2 is a model on the Home page which is made based on the flow in Figure 3.3.1, namely the flow of the SATUSEHAT application which is made in a flowchart to facilitate the creation of the EFSM model in the Altwalker tool. Next, a model will be created for the features that have been selected, and perform a reuse model on the Linked Profile feature.

Based on the naming in table 3.2, the implementation of the reuse model in this study will refer to the Linked Profile model or now in the SATUSEHAT application to be Linked Profile which has been tested previously in Maulidito Dwinandana's research with similar models. Based on the similarity of the model, the template method strategy (template model) and edge template can be used to develop the Linked Profile feature model as done in Guilherme de Cleva Farto's research [5] to produce a model for testing the Linked Profile feature. The following is the Linked Profile model in Maulidito Dwinandana's research which is used as a template method. Figure 3.3.3 is the modeling of the Linked Profile feature on PeduliLindungi which has the same flow based on the model in Figure 3.3.1, but there are slight differences in vertex and edge, such as the start vertex.

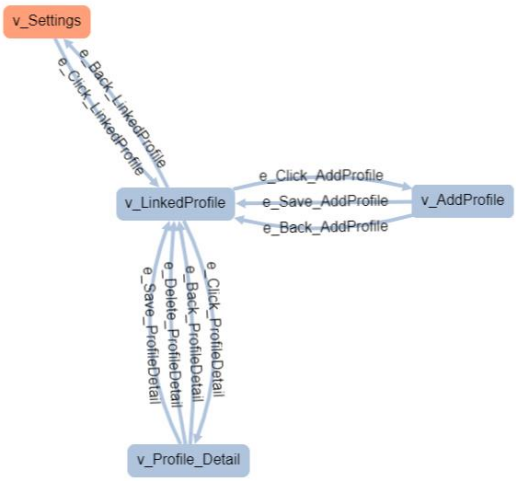

Figure 3.3.3 Modeling Linked Profile features on PeduliLindungi

Figure 3.3.3 is a model of the Linked Profile feature in the PeduliLindungi application. The model becomes a method template to implement the reuse model in this research. The method template became the author's first iteration to develop the Linked Profile feature model by using the edge template as the second iteration, where in the SATUSEHAT application this feature gets additional flow which makes a new vertex and state in the second iteration to adjust to the Linked Profile feature model. Figure 3.3.3 is the design of the reuse model using the template model on the Linked Profile feature.

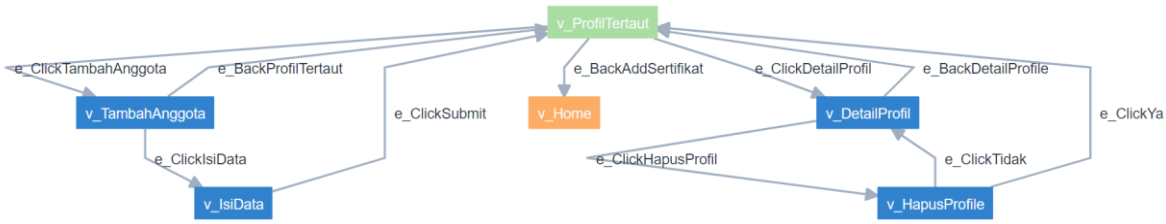

Figure 3.3.4 Example of Modeling Linked Profile features in SATUSEHAT

In Figure 3.3.4 is the second iteration model (edge template), where after the first iteration based on the model in Figure 3.3.3, there are several additional vertex and edge. The first vertex previously named v_Settings changes to v_Profile, because in SATUSEHAT to add members has moved to the Profile page, besides that there is the addition of v_FillData and v_DeleteProfile to adjust to the flow of the SATUSEHAT application.

3.4. Connect Appium

This stage is a preparation stage for executing model-based testing, at this stage the model that has been made will be connected to the application to be tested. Appium is an open-source automation testing tool for applications that is used as a bridge between the code to be created and the emulator as a testing container, to find out the id element on a button in the application Appium is needed. Appium itself, serves to connect the model with the application [14].

3.5. Model Based Testing Execution

At this stage, the author executes Model-Based Testing with Altwalker tools. This execution stage will generate test cases and will directly run the test cases on the Altwalker tools, after the test cases on the tools are run Altwalker will provide results related to the percentage coverage of states, transitions, and requirements. In the execution stage, the author tests sequentially on one feature, where the feature will be executed 3 times to minimize errors during execution.

3.6. Evaluation and Analysis of Results

At this stage, researchers will explain the evaluation and analysis of the results obtained in this study. The researcher will evaluate the results obtained, including the reused model, execution time and coverage level achieved. Furthermore, the analysis is carried out to compare the level of similarity of the template model with the model made by the author, besides that the execution time and coverage level are also compared with previous research, conducted by Maulidito Dwinandana by implementing a reuse model and using different tools to get more effective test results and can solve problems that arise in previous studies.

4. EVALUATION

After implementing the concept of model reuse for testing models in the SATUSEHAT application, the application of method template (model template) and edge template strategies in model reuse can facilitate the process of modeling the same features, because the author uses the model template as the first iteration which is then evolved with the edge template by adding edges to adjust to the flow of the SATUSEHAT application. Using 2 algorithms to compare the reuse model with the template model where the algorithms used are the Weighted_Random algorithm and the Random algorithm. After applying the reuse model, the results of the evolution of the model with the addition of vertex and edge as in table 4.1 are compared with the previous model, there are only 2 features, namely the Linked Profile feature and the Vaccine feature, while the Reminder to Take Medicine feature is a new feature replaced in SATUSEHAT from the Ehac feature in PeduliLindungi.

Table 4.1 Vertex and Edge Comparison

No	Fitur	Perbandingan	Vertex	Edge
1	Profil Tertaut	Model Template	6	11
		Setelah Reuse	8	13
2	Vaksin	Model Template	10	17
		Setelah Reuse	15	24
3	Pengingat Minum Obat	Tanpa Reuse Model	20	33

In Table 4.1 is a comparison of vertices and edges between the template model and after reuse there is model evolution, it can be seen that the Linked Profile feature has an addition of 2 vertices and 2 edges, and the Vaccine feature has a significant addition, which is 5 vertices and 7 edges. If the results of the reuse model are represented in numbers, with calculations combining both aspects (number of vertices and number of edges) and using the average percentage of similarity, the overall similarity percentage value is around 79.81% on the Linked Profile feature. Based on this percentage, it can be concluded that by using the reuse model, in this study the authors only modeled 20.19% of the Linked Profile feature model in the PeduliLindungi application. As for the Reminder to Take Medicine feature, it does not use the previous model due to the change in the name of the Ehac feature. Based on the evolution of the model after model reuse, the level of coverage will affect the success of testing. Table 4.2.1 shows the test results on the SATUSEHAT application with the EFSM model that has been carried out by the reuse model and compares with the results of research conducted by Maulidito Dwinandana.

Table 4.2.1 Test Results on SATUSEHAT Application

No	Model	Feature	Algoritma	Total Vertex	Total Edge	Vertex Coverage	Edge Coverage
1	Model Reuse	Profil Tertaut	Random	8	13	100%	92%
			Weighted Random			100%	100%
	Model Template		Random	11	6	100%	72%
			Weighted Random			100%	54%
2	Model Reuse	Vaksin	Random	15	24	100%	87%
			Weighted Random			100%	83%
	Model Template		Random	17	10	100%	76%
			Weighted Random			64%	34%
3	Model Reuse	Pengingat minum obat	Random	20	33	100%	78%
			Weighted Random	63	31	67%	63%

Table 4.2.1 provides the test results obtained from the application of the reuse model to the EFSM model between PeduliLindungi and SATUSEHAT, where the table compares the results of the reuse model with the template model, using the Altwalker tool and different algorithms are used. Random and Weighted Random are executed three times where the best execution results are taken, and the results are consistent with the data presented in table 4.2 for coverage results. As a result of the test, it can be seen that the Weighted_Random algorithm managed to

achieve 100% vertex coverage, but on edge coverage it was only successful on the Linked Profile feature.

Tabel 4.2.2 Test Report Result on Pass or Failed

No	Fitur	Perbandingan	Vertex	Edge	Status
1	Profil Tertaut	Model Template	6	11	Passed
		Setelah Reuse	8	13	Passed
2	Vaksin	Model Template	10	17	Passed
		Setelah Reuse	15	24	Passed
3	Pengingat Minum Obat	Tanpa Reuse Model	20	33	Passed

Table 4.2.2 provides the test result for pass or failed the features are tested. The results shows that every feature that are tested using the Model Reuse shows Passed which can be concluded that Model Template can be recreate using Model Reuse with the testing results to be improved based on coverage and passed test features.

5. DISCUSSION

In the following research, based on the template model from the research results [3] on the PeduliLindungi application, we found that there was an increase in the coverage rate using the reuse model in the SATUSEHAT application. There is 1 feature that is different from the previous application, namely Medication Reminders, but the other 2 features are the same, namely Linked Profiles and Vaccines, getting better coverage results even reaching a percentage of 100% using the Random algorithm. The following research also compares the results of research [3] based on 2 algorithms Random and Weighted_Random where the results of the Random algorithm are better than the Weighted_Random algorithm. We can conclude that the use of the reuse model in the template model can be said to be better in terms of the percentage of test coverage.

6. CONCLUSIONS

Based on the results of testing and evaluation in this research, it can be concluded that the application of the reuse model concept using template models and edge templates in the SATUSEHAT application provides positive results. It can be seen from the evolution of the model on the Linked Profile and Vaccine features, where there are additional vertices and edges after reuse. Tests on the SATUSEHAT application show a high level of coverage, especially on features that apply the reuse model. The Weighted_Random algorithm achieved 100% vertex coverage on the Linked Profile feature, although edge coverage was only successful on that feature, while the Random algorithm used in the reuse model achieved better results than the template model. These results show that the concept of model reuse can improve the efficiency of model building and achieve a better coverage rate compared to models created from scratch. For comparison, the results of previous research on the PeduliLindungi application show that the reuse model on the Profile feature can achieve 100% coverage with the Weighted_Random algorithm and coverage above 90% with the Random algorithm, while in this study, the resulting model achieves a higher level of coverage. This shows the positive potential of applying the concept of model reuse in improving the quality of testing in evolved applications.

Despite the successful application of the reuse model to the PeduliLindungi application model, there were several failures during execution, due to the random algorithm in the Altwalker tool. The random algorithm makes the model run randomly, not in accordance with the flow of the model that has been mapped. The randomness is also influenced by the shared state in the model so that the movement is erratic. For future research, it is recommended to design a model with minimal shared state to avoid randomness that may occur. Thus, future research can focus more on improving the algorithm and model design to support better execution efficiency

REFERENCE

[1] "PeduliLindungi Resmi Berubah Menjadi SATUSEHAT," 2023. https://promkes.kemkes.go.id/.

[2] González-Aparicio, María Teresa, et al. "A new model for testing CRUD operations in a NoSQL database." *2016 IEEE 30th International Conference on Advanced Information Networking and Applications (AINA)*. IEEE, 2016.

[3] Dwinandana, Maulidito, Rosa Reska Riskiana, and Dana Sulistyo Kusumo. "Extended Finite State Machine-Model based Testing on Mobile Application." *2022 1st International Conference on Software Engineering and Information Technology (ICoSEIT)*. IEEE, 2022.

[4] Apfelbaum, Larry, and John Doyle. "Model based testing." *Software quality week conference*. 1997.

[5] G. De Cleva Farto and A. T. Endo, "Reuse of model-based tests in mobile apps," 2017, doi: 10.1145/3131151.3131160.

[6] Ouriques, Joao Felipe S., Emanuela G. Cartaxo, and Patrícia DL Machado. "Test case prioritization techniques for model-based testing: a replicated study." Software Quality Journal 26 (2018): 1451-1482.

[7] Ibrahim, Maulana Malik, Dana Sulistyo Kusumo, and Rosa Reska Riskiana. "Finite State Machine-Model Based Testing on Website Application." 2022 10th International Conference on Information and Communication Technology (ICoICT). IEEE, 2022.

[8] A. S. Kalaji, R. M. Hierons, and S. Swift, "An integrated search-based approach for automatic testing from extended finite state machine (EFSM) models," *Inf. Softw. Technol.*, vol. 53, no. 12, 2011, doi: 10.1016/j.infsof.2011.06.004.

[9] V. S. Alagar and K. Periyasamy, "Extended Finite State Machine," in *Specification of Software Systems*, London: Springer London, 2011, pp. 105–128.

[10] Villalobos-Arias, Leonardo, et al. "Evaluation of a model-based testing platform for Java applications." IET Software 14.2 (2020): 115-128.

[11] Gurbuz, Havva Gulay, and Bedir Tekinerdogan. "Model-based testing for software safety: a systematic mapping study." Software Quality Journal 26 (2018): 1327-1372.

[12] Veanes, Margus, et al. "Model-based testing of object-oriented reactive systems with Spec Explorer." *Formal Methods and Testing: An Outcome of the FORTEST Network, Revised Selected Papers* (2008): 39-76.

[13] A. Turlea, F. Ipate, and R. Lefticaru, "A Hybrid Test Generation Approach Based on Extended Finite State Machines," in *2016 18th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC)*, Sep. 2016, pp. 173–180, doi: 10.1109/SYNASC.2016.037.

[14] Bringmann, Eckard, and Andreas Krämer. "Model-based testing of automotive systems." *2008 1st international conference on software testing, verification, and validation*. IEEE, 2008.

[15] Pretschner, Alexander, et al. "One evaluation of model-based testing and its automation." *Proceedings of the 27th international conference on Software engineering*. 2005.

[16] M. Rocha, A. Simão, and T. Sousa, "Model-based test case generation from UML sequence diagrams using extended finite state machines," *Softw. Qual. J.*, vol. 29, no. 3, 2021, doi: 10.1007/s11219-020-09531-0.

[17] U. H. Khan *et al.*, "Automating Test Case Generation for Android Applications using Model-based Testing," *Emit. Int. J. Eng. Technol.*, 2022, doi: 10.24003/emitter.v10i1.628.

[18] A. Albahli and A. Andrews, "Model-Based Testing of Smart Home Systems Using EFSM and CEFSM," 2021, doi: 10.1109/CSCI54926.2021.00345.

[19] Altom Consulting, "Altwalker," 2023. https://altwalker.github.io/altwalker/.

[20] "Appium Website," 2012. http://appium.io/introduction.html/..