Automated Test Case Generation from Activity Diagram using Depth-First Search

1st Ignasius Teguh Raharjo Rubiyo
School of Computing
Telkom University
Bandung, Indonesia
teguhraharjorubiyo@student.telkomuniversity.ac.id

2nd Muhammad Johan Alibasa
School of Computing
Telkom University
Bandung, Indonesia
alibasa@telkomuniversity.ac.id

3rd Rosa Reska Riskiana
School of Computing
Telkom University
Bandung, Indonesia
rosareskaa@telkomuniversity.ac.id

***Abstract*—Test cases are essential parts to perform software testing as they play very significant role defining the test steps to work. Test cases can be generated either manually or automatically. Creating test cases manually has several disadvantages, such as higher probability in producing inaccurate results. Furthermore, generating test cases manually is ineffective since manual methods will take longer and require high human efforts. Therefore, generating test cases automatically is more preferable to limit the previous problems. Automated testing can also provide more accurate results than manual testing because it is less prone to human errors encouraging software developers to generate test cases automatically. Test cases can be generated from various sources, including UML diagrams. In this paper, we propose a method to generate test cases automatically by utilising depth-first search algorithms. The UML diagram used in this study is an activity diagram. This paper also provides a method to parse activity diagram that is created using the visual paradigm application that has been converted into an XML format. Our method is able to parse and convert the XML format into graph or tree form. Using our proposed method, a test case search will be carried out using the depth-first search on the tree form to generate test cases. Lastly, we also evaluate our method by comparing the generated test cases with cyclomatic complexity, the method will be performed manually and automatically to validate the results of many scenarios. Our result showed that our method could generate the same scenario (100% coverage) as the manual validation method.**

***Index Terms*—Test case, Depth-First Search, UML Diagrams, Activity Diagram.**

I. Introduction

The development of an application aims to produce a system with quality and is in line with the wishes of the user or the intended purpose. One of the system development processes is testing software, an important and critical phase related to software quality [1]. This stage is very important because it aims to ensure the software meets consumer expectations and avoids defects in the system.

The software can be tested manually, though this approach produces inaccurate results, which is less effective. Automated software testing can provide better and more accurate results. A test cases is a set of rules where test manager verifies that the software being tested can work properly and meet user needs [2]. In carrying out the software testing process, it is necessary to have test cases. Manual execution of test cases can be time-consuming, which is why automation is preferred [3]. Generating test cases is a crucial and fundamental testing process that helps identify errors in a system. It involves creating various test cases to evaluate the system's functionality and ensure it performs as expected [2]. Various methods can be used to generate test cases, including UML (Unified Modeling Language). UML is a general-purpose modelling standard language that aims to visualize, analyze, and document the components of a system in the form of a model or design [2].

In this study, the activity diagram, which is a component of UML, was employed. The activity diagram serves as a visual representation of the sequential flow or control of an activity, aiding in modeling the dynamic aspects of an object or the control flow of an operation. Moreover, this diagram can be utilized as a model for generating test cases [4]. This approach aligns with previous research [5], which also utilized activity diagrams for test case generation. Specifically, this study focuses on generating test case coverage for complex and extensive activity diagrams. To achieve this, a complete combination strategy is employed, aiming to achieve 100% coverage on the ”create a user account” activity.

This study makes several contributions to the field, which include proposing three key elements :(1) a traverse activity diagram method using the modified depth-first search (DFS) for a generating test cases system, (2) a parsing method which can change an XML activity diagram from a visual application paradigm to an activity tree diagram, and (3) an evaluation of the system to generate test cases automatically by calculating cyclomatic complexity for manual validation. The method used to create test cases through an activity diagram involves the use of the depth-first search method, which was inspired by a previous study [1]. This study utilized the depth-first search algorithm in conjunction with activity and sequence diagrams. The findings demonstrated that employing the depth-first search method effectively covered all activities, resulting in the generation of accurate and comprehensive test cases. In addition, there is also a study [6] that uses UML diagrams and depth-first search as a search method to diagnose brain tumor diseases which says that the purpose of using this tree-shaped UML diagram is to assist in the search process carried out until the last root node as the destination result.

The use of depth-first search has many advantages in searching, as described in previous research [7], one of which