Conventional Software Testing Using White Box Method

Muhammad Miftakhul Syaikhuddin[*1]**, Choirul Anam**[2]**, Ade Rizki Rinaldi**[3]**, M El Bahar Conoras**[4]
[1,2,3,4]Universitas AMIKOM Yogyakarta
syaimif.dev@gmail.com[*1], ch.a6.rowi@gmail.com[2], belajarjava2gether@gmail.com[3], elconoras@gmail.com[4]

Abstract

Software development process highly relates to analysis, design, coding, testing and implementation processes. Testing process becomes imperative process to maintain a quality product running well. Testing process can be conducted both for structural and object-oriented software. However, the method utilized for structural and object-oriented software is significantly different. Testing structural program can utilize White Box, Black Box, or Gray testing methods. This study White Box Testing has been employed to test a simple application. The testing process using White Box Testing employs some testing techniques based on path testing consisting of some processes, namely testing independent path, developing flow graph, calculating cyclomatic complexity, and developing graph matrices. Hence, the testing process employing White Box method with basis path testing technique can be executed.

Keywords: *White Box Testing, Basis Path Testing, Flow Graph, Cyclomatic Matrices, Graph Matric*

1. Introduction

Software engineering covers two functions, software as a machine and software as a product. There are some required stages in the process of Systems Development Life Cycle (SDLC), consisting of design, system analysis, coding, testing and implementation. This overall process in SDLC has important and interconnected processes from one to another, started from very well planning stage, system analysis, coding, system testing by ensuring the conformity between the analysis and coding phase to have a properly developed and used software.

In the testing proses utilizing SDLC, there have been some existing methods being able to be employed in the testing process, both for testing structural and object-oriented software. Some possible methods being capable to use in application testing having conventional characteristic are White Box Testing, Black Box Testing and Gray Testing [1]. White Box Testing is a system analysis testing to identify the differences between system requirements with the developed or existing system [1].

The principle of an application test to effectively and efficiently run includes: 1) Testing must be based on user requirements; 2) The available testing time and resource are limited; 3) Resources need to be effectively used for testing; 4) Testing should begin in small and progressive terms in far greater terms; 5) Testing should be implemented by a different team of examiners or an external team; 6) All testing should be based on the customer needs; 7) The best available person should be assigned to do software testing; 8) The test reports including test cases and test reports for a summary of application test results are required; 9) The software testing process should be executed as early as possible in the application development process and should focus on defining the object; 10) The test planning should be completed first; and 11) The initial test plan must and must be updated on time [2].

The quality of the test itself should not be underestimated, there are several criteria in implementing good quality testing, namely having scope of testing for all possible scenarios to operate the software, developing the scope of the paths being as much as possible from the program structure; and being not too simple and complicated [3].

The important steps or method of the testing process using White Box Testing method cover control flow testing, branch testing, base path testing, data flow testing, and loop testing [4]. Meanwhile, according to Pressman [5], there are several testing techniques in White Box Testing

Syaikhuddin, M., Anam, C., Rinaldi, A., & Conoras, M. (2018). Conventional Software Testing Using White Box Method. Kinetik : Game Technology, Information System, Computer Network, Computing, Electronics, and Control, 3(1). http://dx.doi.org/10.22219/kinetik.v3i1.231
Paper submitted on June 14, 2017; Revision on August 15, 2017; Received September 06, 2017

method such as base path testing and control structure testing having some testing methods such as condition testing, data flow testing, and loop testing.

2. Research Method

The research method applied in this research was implemented by doing the process of testing a conventional application or a structural based application used to convert numbers into a sentence. This test is completed by taking sample source code from the application; afterwards, the acquisition process was carried out and determined White Box Testing method as the most suitable technique used in the testing process. Hence, the sample source code will be used to describe the testing process using existing techniques in the White Box Testing method.

White Box Testing method can utilize some techniques in the source code program testing as follows:

1. Loop testing
   Loop testing is one of testing types from White Box Testing having characteristic to focus intesting the validation of iteration construction in a source code [4].
2. Branch Testing
   Brach Testing is also called Statement Coverage and Branch Coverage, a method used to validate every code line executed at least once [6]. The branch in programming language such as ‘IF Statement’ has two values namely true and false.
3. Basis Path Testing
   Basis Path Testing explained by Gupta [7] referred to McCabe ensures all independent code paths have been tested. An independent path passing every code line introduces at least one new set, a new statement or new condition.

White Box Testing has considered as a seldom utilized testing method, but it will be more preferable to be employed to test in specific condition. This condition needs a tester to review a code program having high complexity level [8]. These techniques are a variation of White Box Testing technique. Figure 1 presents more details on this connection.

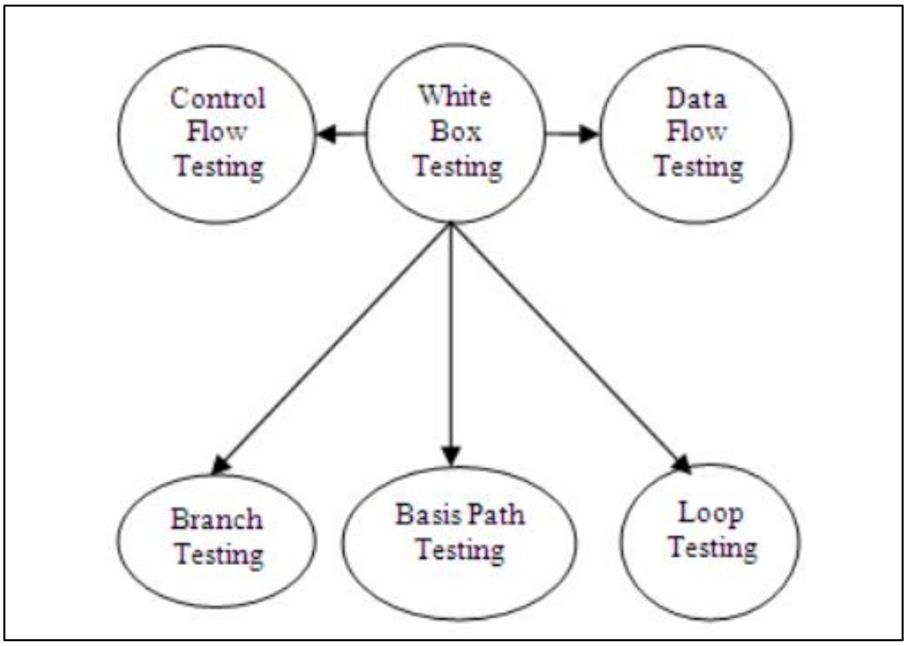


Figure 1. Representation of Technique Variation in White Box Method

The utilized research flow as described in the first paragraph of the research methodology chapter covers some processes initiated by collecting related literatures to have better understanding to the description of the technique. The following phase is applying the source code sample into the techniques used to clarify the testing process, used as a proofing step in the test technique. Furthermore, it has some testing methods that can be used to obtain more accurate proof. Figure 2 shows more details on the research flow.

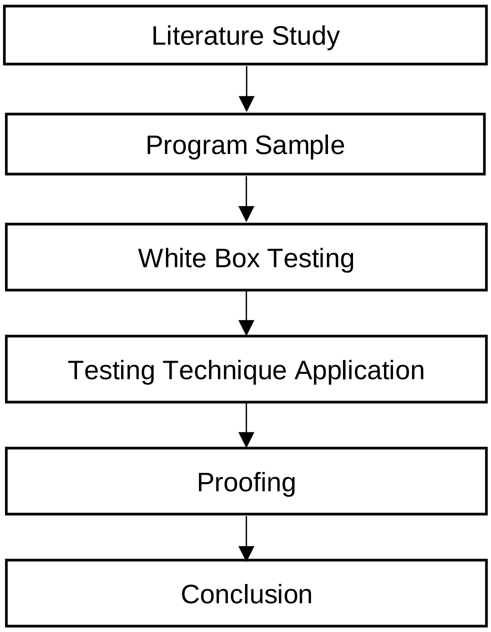


Figure 2. Research Flow

The testing technique used in testing by White Box Testing method is basis path testing. Basis path testing utilizes several steps to complete its testing process, flow graph notation, cyclomatic complexity, deriving test case and graph matrices [4].

3. Research Results and Discussion

The discussion on this research covers the utilization of White Box Testing and basis path testing. Basis path testing is a testing technique in White Box Testing method initially proposed by McCabe and allowing a test case designer to measure logic complexity from procedural design. In addition, this measurement is used as an approach to decipher the execution of a basis path set (a basis set is a set of all implementation procedures) [4]. However, there some other opinions related to basis path testing as a technique utilized to test a source code program based on flow control. This approach uses control bot flow to alter code into model generating independent paths [1]. Basis path testing technique employs some approaches, flow graph notation, cyclomatic complexity, graph matrices, and deriving test case.

3.1 Flow Graph Notation

Flow Graph Notation is structural testing strategy using a path control program as a path control model, having a simple approach with a complex path. Control Flow Testing is applicable to almost all software and the most effective testing to most software; moreover, this testing becomes the most basic technique in White Box Testing, and its application is mostly designated to smaller application programs or segments of bigger application program [6].

However, there are some existing opinions stating simple notation used to represent control flow is also by flow control. Flow graph is a logical control flow utilized to illustrate control program structure. It is important to notice that every circle in the flow graph is called by flow graph node; the arrow is called by edge or link (representing control flow); and the area confined by a node and edge is called by a region. The following flow graph represents an individual construction combined to construct a flow graph for a specific procedure [4].

3.2 Cyclomatic Complexity

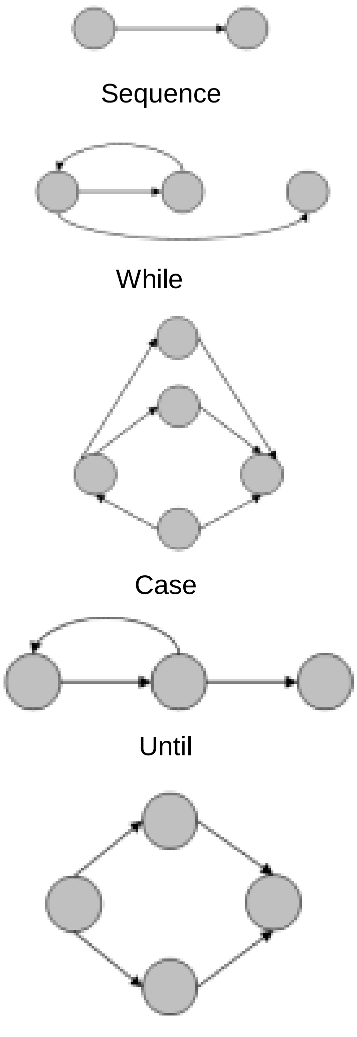
Cyclomatic complexity is software matric providing the fourth quantitative measurement from logical complexity. If used in basis path testing, this technique defines the number of independent paths (any passing paths in a program having at least a new set or a set of processing statement or a new condition) in the basis program series providing high value for a number of required tests to ensure overall coverage of program statements [4].

In addition, other opinions state cyclomatic complexity is the preferably used matric complexity in software engineering, defined by Thomas McCabe in 1976 based on control path structure of a program. Hence, it eases in understanding, calculating and providing useful outputs [2].

There are some steps in calculating cyclomatic complexity in a program, namely:

1. Cyclomatic Complexity: V(G), the flow graph G defined as:
   V(G) = P + 1
   P represents the number of predicate nodes.
2. Calculating the number of regions in a flow graph according to cyclomatic complexity.
3. Cyclomatic Complexity: V(G), the flow graph G defined as:
   V(G) = E-N+2
   N represents the number of nodes (points in a flow graph), and E represents the number of edges.

Sub chapter 3.5 will discuss these formulations.



Sequence

While

Case

Until

If

Figure 3. Illustration of Flow Graph Types

3.3 Deriving Test Cases

Obtaining a test case of a procedure becomes the main purpose from basis path testing. Some required phases in basis path testing are as follows [4]:

1. Developing a suitable flow graph according to proposed design or source code.
2. Calculating the cyclomatic complexity from the flow graph using from one the given formulas.

3. Determining one linear path in the independent paths.
4. Preparing test case used to execute every path in the basis set.
5. Implementing testing process at least once for every testing path.

3.4 Graph Matrices

Graph Matrices are two-dimensional matrices being very helpful in determining basis set, having similar number of columns and paths with the existing nodes in a flow graph. In addition, letters are usually assigned in order to differentiate each node. Each edge is completed with some links (“0” value = no connection and “1” value = connection) [4].

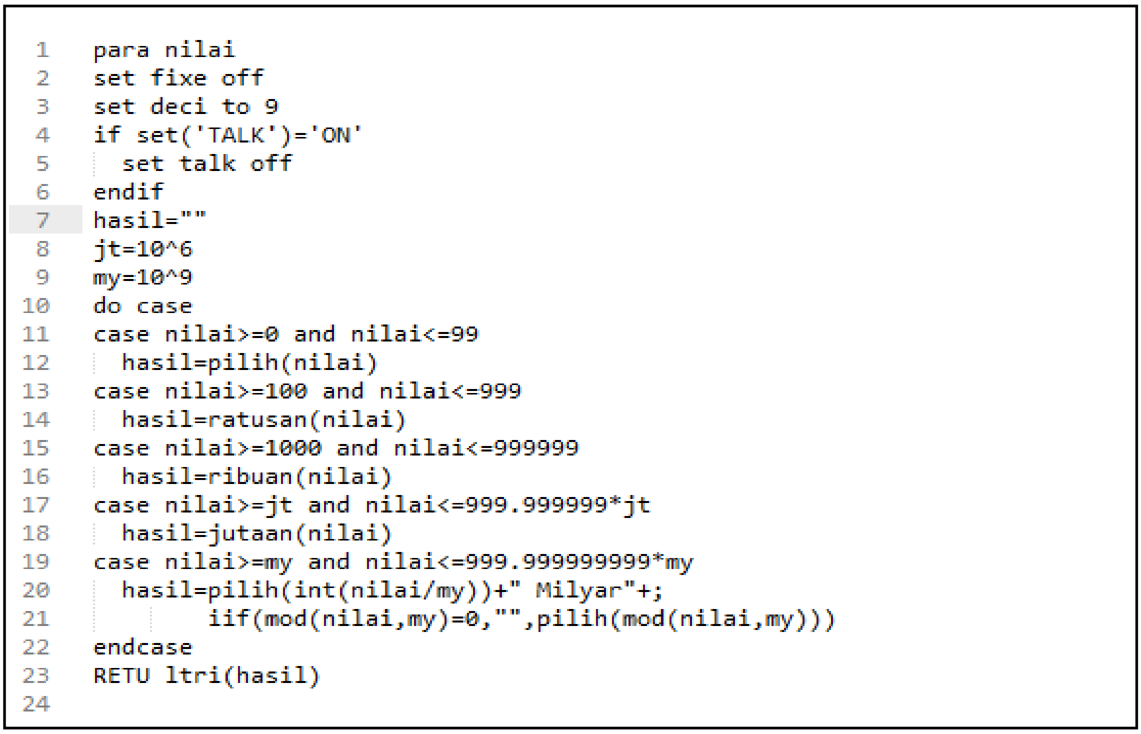
This research implemented testing by converting the flow graph to square matrix employing one line and column for every graph. Therefore, tracking all links in the flow graph having been passed at least once could be implemented [2].

3.5 Example of Cyclomatic Complexity and Graph Matric

To strengthen the previous review, the research performed a trial using provided source code in the form of simple application module being capable to convert numbers to sentences or words. Furthermore, this trial utilizing basis path testing technique with cyclomatic complexity and graph matrices.

3.5.1 Cyclomatic Complexity Testing

The initial stage was reading and understanding the source code in order to construct a flow graph as this trial’s most important process, leading to the further processes.



```
para nilai
set fixe off
set deci to 9
if set('TALK')='ON'
  set talk off
endif
hasil=""
jt=10^6
my=10^9
do case
case nilai>=0 and nilai<=99
  hasil=pilih(nilai)
case nilai>=100 and nilai<=999
  hasil=ratusan(nilai)
case nilai>=1000 and nilai<=999999
  hasil=ribuan(nilai)
case nilai>=jt and nilai<=999.999999*jt
  hasil=jutaan(nilai)
case nilai>=my and nilai<=999.999999999*my
  hasil=pilih(int(nilai/my))+" Milyar"+;
        iif(mod(nilai,my)=0,"",pilih(mod(nilai,my)))
endcase
RETU ltri(hasil)
```

Figure 4. Source Code of the Converter Application

From the source code presented in Figure 4, a flow graph culd be generated based on statements in the source code. As mentioned in Figure 5, there are 10 statements according to the given numbering. The flow graph from the previous flow graph is presented in the following figure.

From the flow graph, the process to determine regions in cyclomatic complexity could be conducted by searching the existing independent path in the flow graph. Those paths are as follows:
Path 1: 1-10
Path 2: 1-9-1-10

Path 3: 1-2-3-4-9-1-10
Path 4: 1-2-3-5-9-1-10
Path 5: 1-2-3-6-9-1-10
Path 6: 1-2-3-7-9-1-10
Path 7: 1-2-3-8-9-1-10
Calculating cyclomatic complexity from the flow graph, subsequently, could be performed as:

R, Region = 7
Number of node = 10
Number of edge = 15
Number of predicated node = 6

This obtained data would be applied to the related formula to obtain:
Cyclomatic Complexity
V(G) = R = 7
or
V(G) = Predicate node +1
= 6 + 1
= 7
or
V(G) = Edge – Node + 2
= 15 – 10 + 2
= 7

Therefore, the cyclomatic testing equals to 7.

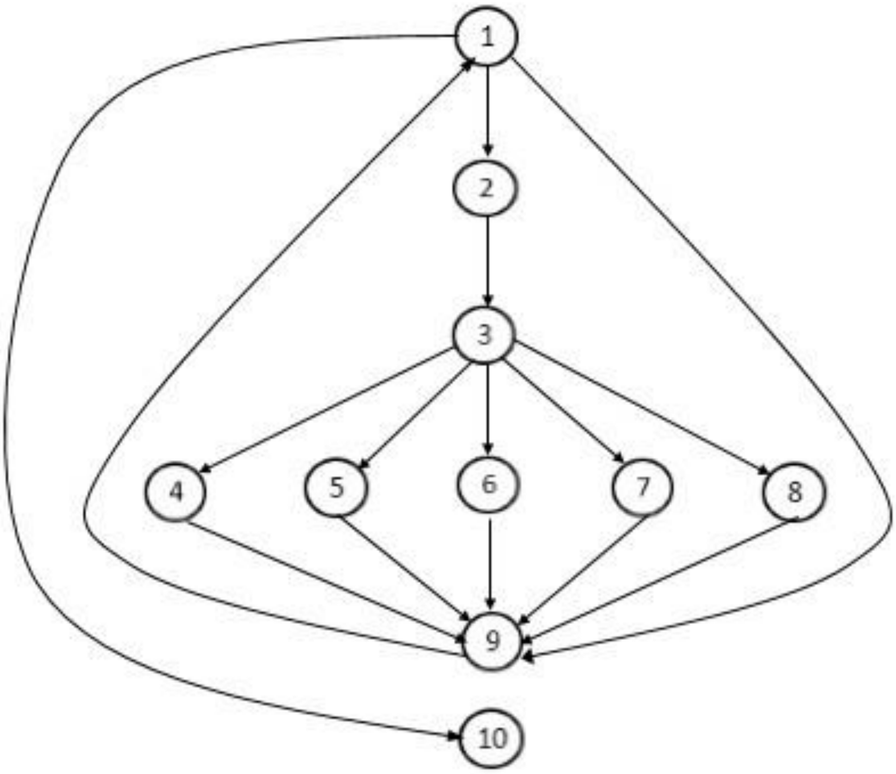


Figure 5. Example of Source Code Flow Graph

3.5.2 Graph Matric

After calculating using cyclomatic complexity, there is an alternative method by using graph matric from the existing flow graph. This scenario is presented in Figure 6.

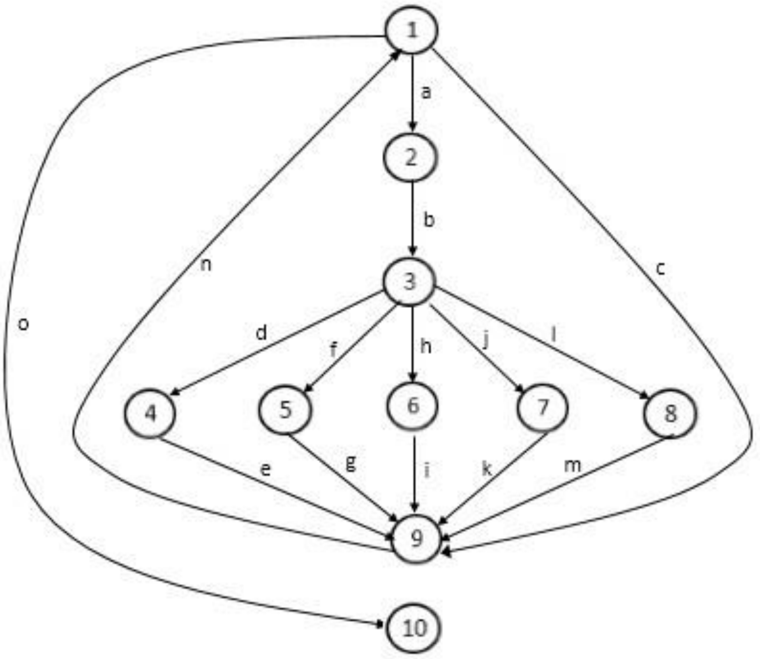


Figure 6. Graph Matric

From the given graph matric, a graph matric Table 1 and a connection matric Table 2 can be developed.

Table 1. Graph Matric

Node	1	2	3	4	5	6	7	8	9	10
1		a							c	O
2			B							
3				D	F	h	j	l		
4									e	
5									g	
6									i	
7									k	
8									m	
9	n									
10										

Moreover, the table for the connection matric is as follows:

Table 2. Connection Matric

Node	1	2	3	4	5	6	7	8	9	10	Connections
1		1							1	1	=3-1=2
2			1								=1-1
3				1	1	1	1	1			=5-1=4
4									1		=1-1
5									1		=1-1
6									1		=1-1
7									1		=1-1
8									1		=1-1
9	1										=1-1
10											=0

Thus, the cyclomatic complexity is calculated as 2+4=6.

4. Conclusion

The whole completed testing process presents some difference results on each chosen technique of the testing process. The cyclomatic testing result shows 7; on the other hand, the graph matric testing indicates 6. It may be caused by an error in the flow graph construction or in the source code development. Consequently, it will be necessary to find an experienced tester to conduct the testing process using White Box Testing method due to its complexity.

Compared to Black Box Testing by conducting testing only on the existing system function, White Box testing is used to test the procedure in its source code program. Additionally, there some benefits in using White Box Testing exemplified by the availability to execute all logical decisions and all independent paths in a model [9]. An automatic testing process to minimize the software's production cost and improve the software's reliability can also be executed [10].

References

[1] Rathod SS. "*A Scenario of Different Types of Testing Techniques in Software Engineering*," International Journal of Advanced Research Computer Science and Software Engineering, Pp. 133–44, 2014.

[2] Dhingra N, and Mayank, "*Contingent Study of Black Box and White Box Testing Techniques*," International Journal of Current Engineering Technology, Pp. 3346–52, 2014.

[3] Rouf A., "*Software Testing using White Box and Black Box Testing,"* Journal of Information Technolog*y HIMSYA-Tech*, Pp. 1–7, 2012.

[4] Khan ME., "*Different Approaches to White Box Testing Technique for Finding Errors*," International Journal of Software Engineering and its Application," Pp. 1–14, 2011.

[5] Pressman RS., "*Software Engineering: A Practitioner's Approach*," 7th Ed, 2009.

[6] Kumar M, Singh SK, and Dwivedi RK., "*A Comparative Study of Black Box Testing and White Box Testing Techniques*," International Journal of Advanced Resesearch in Computer Science and Management Studies. Pp. 32–44, 2015.

[7] Gupta N., "*Different Approaches to White Box Testing to Find Bug*," International Journal of Advanced Research in Computer Sciency and Technology (IJARCST 2014), ;2(3):46–9, 2014

[8] Dennis A, Wixom BH, and Roth RM., "*System Analysis and Design*," 5th Edition. Hoboken: John Wiley & Sons, Inc., Pp. 594, 2012.

[9] Jovanovic I., "*Software Testing Methods and Techniques*," IPSI BgD Transaction on Internet Research," [Internet], Pp. 30–41, 2009. Available from: http://www.internetjournals.net/journals/tir/2009/January/Full Journal.pdf#page=31

[10] Shao D, and Perry DE., "*Whispec: White Box Testing of Libraries using Declarative Specifications*," Framework, Pp. 11–20, 2007.