LJISSEN-AI

Lovelace Journal of Information System, Security, Education and Network Artificial Intelligence

E-ISSN: XXXX-XXXX
P-ISSN: XXXX-XXXX

*Article*

# Implementation of Path Testing in White Box Testing for the National Public Transportation Driver Behavior and Competency Assessment System

**Safriza[1], Hetty Rohayani[2]**

[1] Prodi Sistem Informasi, Univesitas Pembangunan Jaya ; e-mail : safrizal.Abdurrahman.@upj.ac.id
[2] Prodi Informatika,Universitas Muhammadiyah Jambi e-mail : hettyrohayani@gmail.com

**Abstract:**The National Public Transportation Driver Behavior and Competency Assessment System is a system designed to assess driver behavior and competence in order to improve the quality of service and safety of public transportation in Indonesia. The main problem in this study is the less than optimal testing of the program logic as a whole which has the potential to cause system errors in providing assessment results. Therefore, this study aims to apply the Path Testing method in White Box Testing to evaluate and ensure that all logic paths in the system have been completely tested and produce output as expected. The method used is Path Testing, which is a code-based testing approach that focuses on identifying and testing all independent paths in the program control structure. This process is carried out by analyzing the flowchart and control flow graph (CFG) of the system being tested, then designing and executing test cases based on the paths obtained. The test results show that the Path Testing approach is able to reveal program paths that were previously undetected by ordinary functional testing. This test successfully ensures that each branch of the system logic works correctly according to specifications. These findings confirm that the use of the Path Testing-based White Box Testing technique is very effective in improving system reliability and accuracy. The conclusion of this study is that the systematic application of Path Testing is able to improve software quality, prevent logical errors in the assessment process, and guarantee objective, accurate evaluation results in the national public transportation driver assessment system.

**Keywords:** White Box Testing; Path Testing; Assessment System; Driver Competence; Software Testing.

## 1. Introduction

This study is used to assess the behavior and competence of public transportation drivers, such as bus and public transportation drivers, at the national level. The assessment of the behavior and competence of public transportation drivers is adopted using the criteria in the Graduate Competency Standards (SKL) based on the Indonesian National Qualification Framework (KKNI) regarding level II beginner drivers. The assessment instrument is based on the Graduate Competency Standards (SKL). SKL has 4 competency components, the 4 competency components include attitudes and values, abilities in the field of work, knowledge mastered, rights and responsibilities. [1]

This study uses White Box testing, namely path Testing. The methodology is by conducting Control Flow Graph (CFG) Analysis, namely by compiling CFG from the source code of each main function, analyzing the program logic flow such as branching, repetition, and conditions. Determining the Determination of the Basis Path and Independent Path by calculating Cyclomatic Complexity to determine the number of independent paths. Designing and Executing Test Cases by Compiling test cases based on the specified logic path. [2], [3]

The disadvantage of Path testing on White Box is that it requires a deep understanding of the internal structure of the code and the logic of the program. Testing the paths that exist in the code, logical paths that should exist but will not be detected if not implemented (Katalon. (n.d.).

The advantage of path testing in White Box testing is that it detects logical errors comprehensively, checks all execution paths, so that it can find logical errors that are not detected

DOI : https://doi.org/...........

by other methods. [1] Improves understanding of algorithm design and program logic Because it requires an understanding of the structure in the code, path testing helps developers understand the logic of the system in depth. [2] The Problem of White Box Testing in the Application of Path Testing to the National Public Transportation Driver Behavior and Competency Assessment System is the difficulty in identifying Independent Paths Path testing requires identification of path bases or independent paths through control flow graphs (CFG). However, the assessment system is based on many conditions and loops (eg: assessment based on time, violation history, all paths accurately [3] The approach used is white box testing, especially path testing. By identifying the basis path or independent path using control flow graph (CFG). Determining independent paths manually, ensures that all paths have been tested [4] The contribution of this study is the implementation of path testing, based on the control flow graph (CFG) independent paths are identified, so that they can be used as a standard in system testing. [1] Gap analysis of this study, many studies only use Black Box Testing or ordinary functional testing for transportation systems. Transportation system testing often focuses on input-output results without testing the internal logic structure of the program. Lack of application of White Box Testing based on Path Testing in the driver behavior assessment system. This study uses Cyclomatic Complexity calculations to determine the number of independent paths and test them all thoroughly - something that is rarely discussed in the implementation of human behavior system testing. [5]

## 2. Preliminaries or Related Work or Literature Review

White box testing is performed at the unit, integration, and system levels to detect logical errors, debug, and validate programming assumptions. This method includes thorough testing of all code paths, helps with optimization, and provides guidelines for stopping testing. However, this testing is complex, requires high expertise, can be expensive, and is less practical for large applications. Therefore, it is recommended to be performed at the unit testing phase in the software development cycle. [1] Path Testing to improve system reliability is one of the techniques in white box testing that focuses on testing all logical execution paths in the program code. Because every part of the code logic is tested, potential bugs or failures can be found and fixed before the system is used. This will result in a more stable, accurate, and reliable system. [2] White box testing is a software testing method that involves a deep understanding of the internal structure of the program. One of the techniques used in White Box Testing is Basic Path Testing, which focuses on testing every possible execution path in the program. This testing aims to ensure that the program functions properly and can manage the lending process well ([3] White-box testing is an important component of the automated development process carried out in the Continuous Integration (CI) or Continuous Delivery (CD) pipeline. The white-box testing methods used are Branch Testing and Basis Path Testing. White-box testing is done by analyzing the application source code first, determining the files and functions that must be tested, and creating a flow graph from the source code ([4] White box testing involves access to the source code and is carried out during software development. The test results generated from white box testing depend on the flowchart that is transformed into a flowgraph. The white box testing method with the basic path technique produces test results that are increasingly large along with the size of the system being tested [5] White box testing is also known as structural testing, where the tester has access to the entire structure or logic of the software being tested (Selfia & Arimby, n.d.). The steps used for path testing are Flowchart/Sourcecode Analysis, create Flow Graph, Cyclomatic Complexity, Determining Independent Paths, Designing Test Cases. Cyclomatic Complexity is a measure of the number of different execution paths in a program. The higher the number, the more complex the code, and the harder it is to test and maintain. $V(G) = E – N + 2$.

Description: E = number of edges in the flowgraph N = number of nodes in the flowgraph, P = number of predicates nodes in the flowgraph [6] This study uses White Box testing to use several testing techniques based on path testing consisting of several processes, namely independent path testing, flow graph development, cyclomatic complexity calculations, and graph matrix development. [7]. The white box testing method prioritizes knowledge of the system structure. In real practice, this method can be applied in testing information system products to ensure their reliability and quality [8]. Software testing is one aspect of the software design method. Research by taking a case study on the inventory information system of goods in a goods delivery service business. This information system testing technique uses the black box testing method, with the Equivalence Partitioning testing technique. [9] White box testing requires an in-depth understanding of the internal structure of the software and the programming language used. This makes it less suitable for testing teams that do not have adequate technical knowledge or access to the source code. [10]

White Box Testing functions as Clear Box Testing or Structural Testing to assess the internal structure of the application and logical functions and programming code. [11]

The strength of White-Box Testing can be ensured that all code paths, branches, loops, and conditional statements have been covered as identification for hidden errors. Helps detect Early Bugs in Code to find bugs and security vulnerabilities early in the code, which cannot be done in black-box testing. [12]

Web application testing is a software practice designed to ensure quality by verifying that the functionality of a particular web application is functioning correctly or in accordance with the specified requirements. Web testing makes it possible to find bugs at any time, before release or every day. [13]

The problem with this software development is often the lack of testing that causes software failure. To maintain high-quality products in excellent performance conditions, testing becomes important. Software can be tested using White Box, Black Box, or Gray testing techniques. [14]

This study aims to produce a method for optimizing white box testing on Java programs by utilizing branching and repetition structures using the basis path method, and analyzing the effectiveness of the proposed method in generating test cases. [15]

White box testing methods are tailored for transformer-based NLP models. These methods include Mask Neuron Coverage (Mncover) which measures how thoroughly the attention layers in the model are exercised during testing. It further shows how Mncover can be used to guide the creation of Checklist inputs, evaluate alternative NLP testing methods, and encourage data augmentation to improve accuracy.[16]

Two methodologies are used in both conventional and object-oriented approaches: White box testing and black box testing. White box testing focuses on the control structures in the program, to verify all statements in the program are executed at least once and all logical conditions are executed. It includes basic path testing, condition and data flow testing, program logic, and loop testing. Unlike white box testing, black box testing does not require knowledge of the internal workings of the program. Testing is done to verify functional requirements. It includes equivalence partitioning, boundary value analysis, error-based testing, random testing, partition testing, cause-and-effect graph techniques.[17]

This research aims to equip these algorithms with explainable AI (XAI) features to improve transparency. Testing this methodology on three different medical datasets reveals consistent correlations between the adapted White box models and their black box counterparts. Notably, integrating this strategy with established methods such as locally interpretable model-agnostic explanations (LIME) and SHapley Additive explanations (SHAP) further enhances transparency, underscoring the potential value of decision trees as the preferred White box algorithm. in medicine because of its inherent explanatory power. These findings highlight a promising path for integrating black-box algorithm performance with the need for transparency in critical decision-making domains.[18]

Testing can never completely identify all defects in software. Instead, testing provides a warning or evaluation that compares the state and behavior of the product against principles or mechanisms that one might use to recognize a problem. Software testing is an important term for software reliability. Testing provides the original structure and validity to the software for efficient performance under operational conditions.[19] Automated white-box testing is a major problem in software engineering.

Over the years, several tools have been proposed to support different parts of the testing process. However, these tools are mostly separate and most of them support only a small set of fixed and limited test criteria. In this paper, FRAMA-C/LTEST, a generic and integrated tool for automated white-box testing of C programs, is presented. LTEST provides integrated support for various test criteria as well as easy integration of new criteria. [20]

Evolutionary white-box testing is a promising approach for the complete automation of structure-oriented test case generation. Evolutionary testing cannot find valid test data. The termination criteria of evolutionary white-box testing can be tailored to the test objectives with problem cases in such a way that problematic test objectives are excluded from testing in advance or can be covered due to adequate termination criteria according to the software size. This can lead to increased efficiency and effectiveness of evolutionary white-box testing. [21]

Evolutionary white-box testing has so far not been able to make statements about the existence of unattainable test goals, which can be improved with the help of evolutionary software measurements. This improvement is based on the ability of evolutionary software measurement to estimate in advance the effort To implement software measurement requires the calculation of dynamic stopping criteria, it is also necessary to examine the distribution of the actual measured test effort compared to the effort estimated by software measurement. [22]. This study shows that this method successfully identifies and fixes most errors in programming logic and system vulnerabilities, which contributes to a 20% increase in performance, increased stability, and reduced risk to security threats. [23]. White box testing has a fairly popular technique, namely the basis path. White box testing with the basis path technique is applied to the population administration system website, which is a computer application for carrying out population administration. Testing is carried out on the features on the RT account after testing is carried out by creating a flowchart of the features to be tested, then creating a flowchart according to the source code, and testing the source code, after that calculating the entire scenario to be tested using Cyclomatic Complexity (CC), and comparing the results of the tests carried out with the expected results. It was obtained that the path of each feature that had been passed was in accordance between the results of the CC and the Flowgraph created. [24]

Testing is the process of running a program to identify errors, need to improve the accuracy of the coverage area by including data flow elements based on aliases and avoiding redundant elements that reduce the overall coverage to improve the applicability and effectiveness of data flow testing. This test describes the steps to perform data flow testing as well as how to design a test suite that accounts for anomalies. Examines and discusses the methods used to date to perform data flow testing. These approaches include node-based design, trend discovery coverage, web application comparison, and analytical testing.[25]

## 3. Proposed Method

Implementation of Path Testing in White Box Testing for the National Public Transportation Driver Behavior and Competency Assessment System. The assessment is carried out by passengers, supervisors, trainers, supervisors and e-learning. From the assessment carried out by passengers, supervisors, trainers and e-learning are entered into the system that has been created. From the assessment system, white box testing is carried out using Path testing. The methodology used is using the Software Testing Life Cycle (STLC) and the White Box Testing Stages. Regarding STLC and the White Box Testing Stages are explained as follows:

Software Testing Life Cycle (STLC) is a systematic process consisting of several stages carried out to ensure the quality and reliability of software through structured testing. STLC not only includes the testing process itself, but also the planning, preparation, implementation, and evaluation of test results.

STLC is a series of stages carried out in the software testing process, starting from test planning to test closing, with the aim of finding and fixing bugs, and ensuring that the software works according to the expected specifications.

Stages in STLC:

1. Requirement Analysis (Requirement Analysis)
   The QA (Quality Assurance) team analyzes the requirement document to understand what should be tested with the objectives: identifying test needs and test feasibility.
2. Test Planning (Test Planning)
   Develop a test strategy, determine resources, budget, schedule, and tools to be used. With the main output of the Test Plan or Test Strategy Document.
3. Test Case Development (Test Case Development)
   Write test cases and test scripts based on functional and non-functional requirements, including test data preparation.
4. Test Environment Setup (Test Environment Setup)

Prepare the environment needed to run the test, such as servers, databases, and supporting software.

5. Test Execution
   Running test cases and recording the results. Bugs or errors found will be reported using a bug tracking tool.
6. Test Cycle Closure
   Evaluating the entire test cycle, documenting the final results, lessons learned, and submitting a final test report.

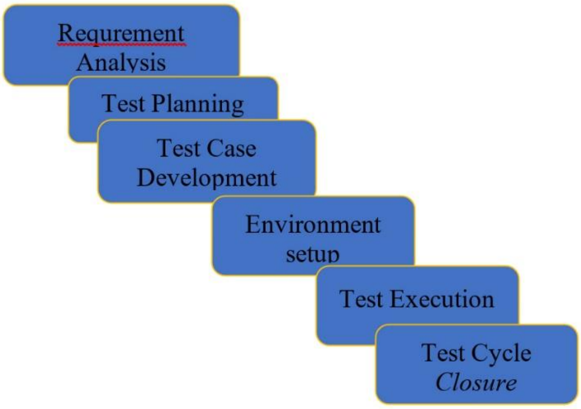


**Figure 1** ***Software Testing Life Cycle (STLC)***

### White Box Testing Stages

The testing stages consist of flowchart/sourcecode analysis, creating flow graphs, cyclomatic complexity, determining independent paths, and designing test cases as shown in Figure 2.

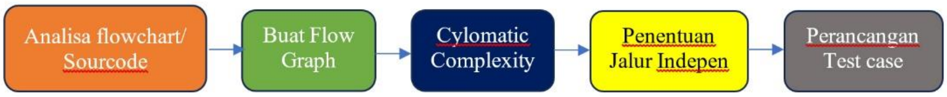


Figure 2. White Box Testing Stages

Explanation of figure 2 is as follows:

Flowchart/Sourcecode analysis, is a graphical representation of a process or logical flow. In software testing, flowcharts are used to: Visualize business processes or system logic. useful for understanding how the program flow runs, determining critical points for testing (decision points, loops, input-output).

2. Flow Graph is a directed graph that describes the flow of program execution. Nodes represent blocks of instructions, and edges indicate the direction of program execution.

3. Cyclomatic Complexity is a quantitative measure of the number of independent execution paths in a program or module. Introduced by Thomas McCabe in 1976, this metric is used to: Assess the logical complexity of code, Determine the minimum number of test cases required for full path coverage. Assist in the process of code refactoring and code maintenance.

4. An independent path is a path in the program control flow (flow graph) that passes through at least one new edge (new direction) that has not been passed by other paths. Represents a unique combination of branching conditions, loops, or

other logical structures. The goal is to: identify all major logical paths in the code. Design minimal test cases that touch the entire program logic.

5. Designing Test Cases

After determining the independent paths from the flow graph above, the next step is to create a test case for at least the number of independent paths that have been created, ensuring that a test case is created for every possible path that will be passed.

## 4. Results and Discussion

In the results and discussion of this research, what is discussed is the use case of Supervisor and Trainer Assessment in Figure 3.

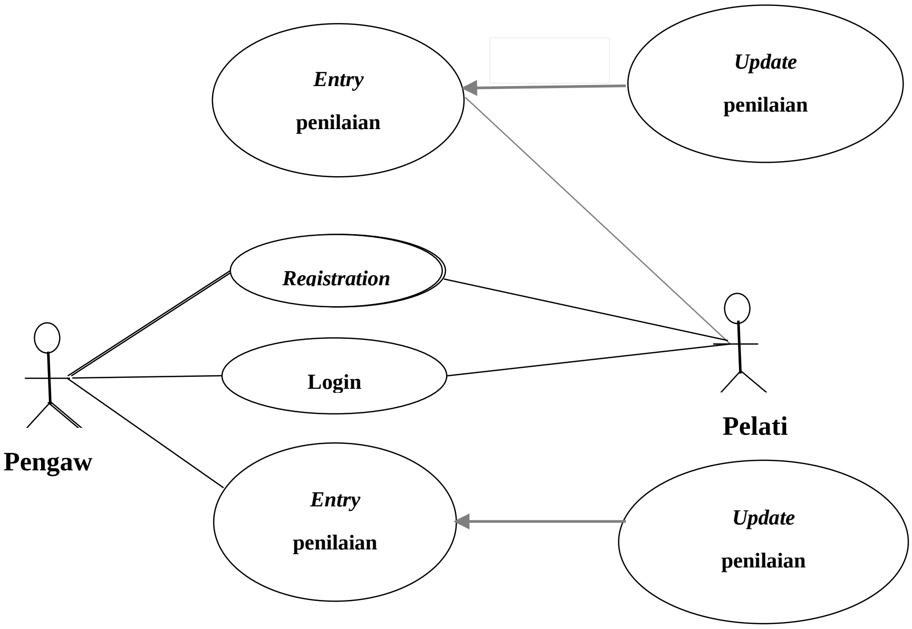


**Figure 3 Use Case Supervisor and Trainer Assessment**

Table 1 explains the description of the image Figure 3 Use Case Supervisor and Trainer Assessment

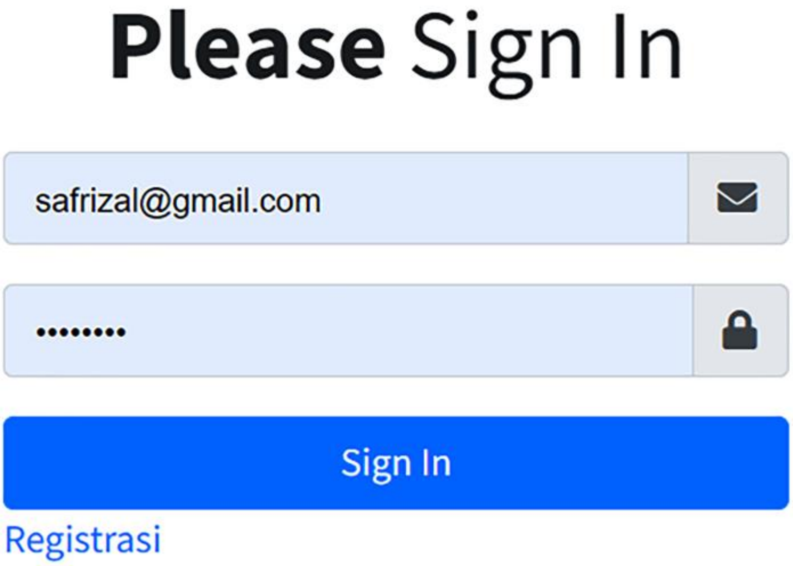
Table 1. Description of Figure 3 Use Case Supervisor and Trainer Assessment

| Main Scenario | |
|---|---|
| Initial Conditions | The actor has logged in to the driver's website |
| Actor Action | System Reaction |
| | The system responds by displaying the main menu. |
| Final Condition | Actors can access the main menu |
| ***Use Case Login-*** Supervisor and Coach Assessment | |
| Objective | login |
| Deskripsi | Supervisors and trainers log in, if they have not registered then register. |
| Actor | Supervisor and Coach Assessment |
| Main Scenario | |
| Initial Conditions | The actor has logged in to the driver's website |
| Actor Action | System Reaction |
| | Sistem The system responds by displaying the main menu. |
| Final Condition | Actors can access the main menu |

For the Application, the menu used in the driver system application system is explained as follows:

1. Login Display

The Login Display for the Transportation System is in Figure 4.



Please Sign In

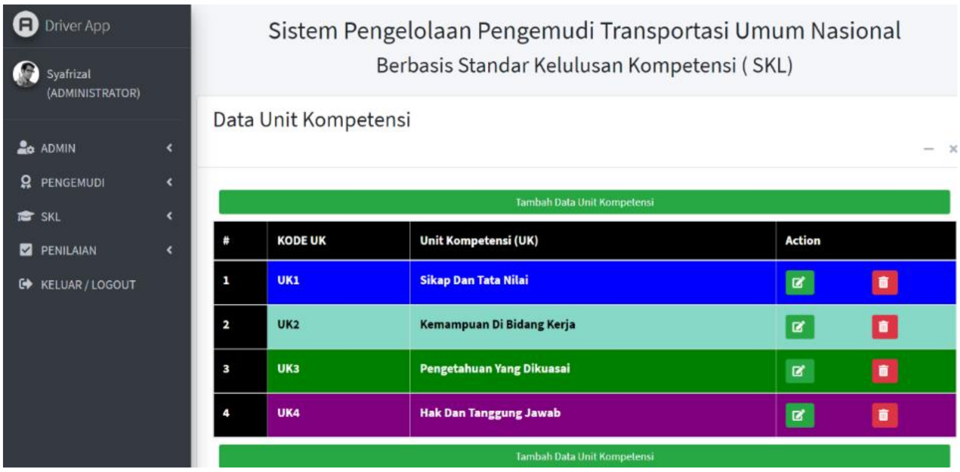
safrizal@gmail.com

••••••••

Figure 4 Login View

2. Dashboard Display

The main menu consists of administrator, driver, assessment, exit/logout as shown in figure 5.



Sistem Pengelolaan Pengemudi Transportasi Umum Nasional

Berbasis Standar Kelulusan Kompetensi ( SKL)

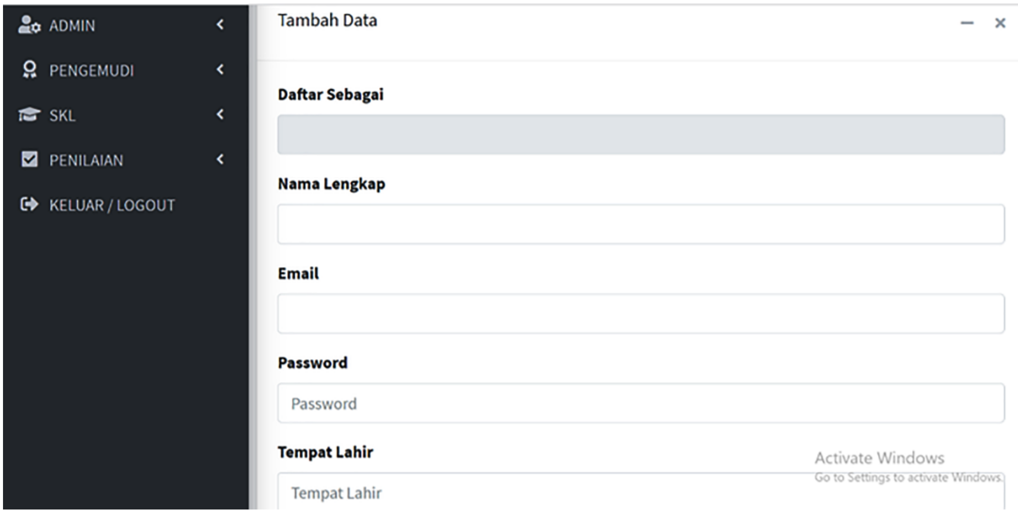
Data Unit Kompetensi

| # | KODE UK | Unit Kompetensi (UK) | Action |
|---|---|---|---|
| 1 | UK1 | Sikap Dan Tata Nilai | |
| 2 | UK2 | Kemampuan Di Bidang Kerja | |
| 3 | UK3 | Pengetahuan Yang Dikuasai | |
| 4 | UK4 | Hak Dan Tanggung Jawab | |

**Figure 5 Main Menu**

3 Driver Profile

Figure 6 Driver Profile, if the driver is clicked it displays the driver profile which contains driver data such as name, email, date of birth, address.



Tambah Data

**Daftar Sebagai**

**Nama Lengkap**

**Email**

**Password**

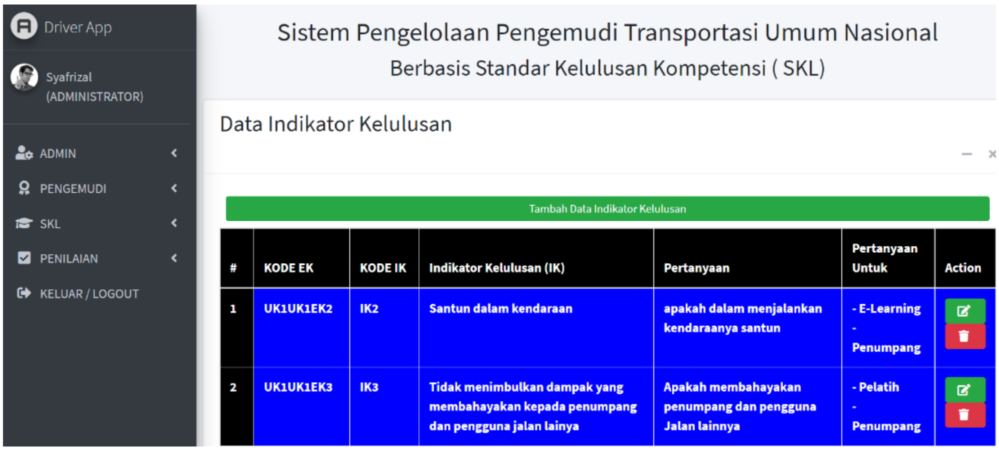
Password

**Tempat Lahir**

Tempat Lahir

## Figure 6 Driver Profile

4. Passing Indicator
Figure 5 illustrates the passing indicator. The passing indicator contains the passing indicator for the driver.



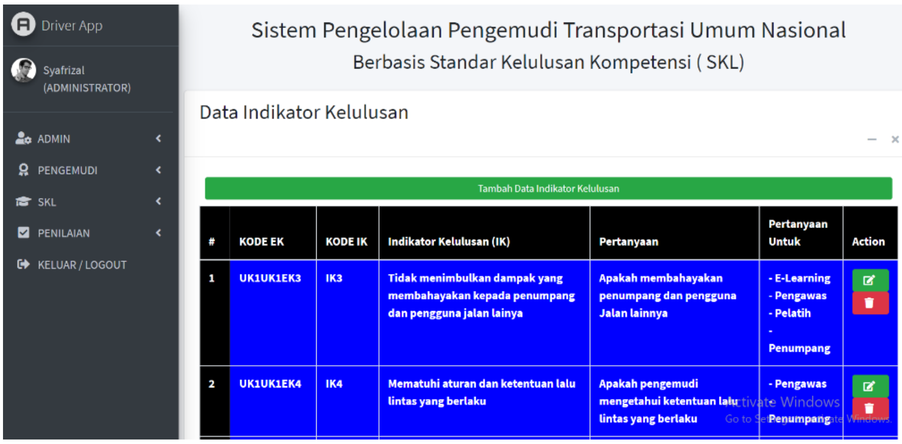
Sistem Pengelolaan Pengemudi Transportasi Umum Nasional
Berbasis Standar Kelulusan Kompetensi ( SKL)

Data Indikator Kelulusan

| # | KODE EK | KODE IK | Indikator Kelulusan (IK) | Pertanyaan | Pertanyaan Untuk | Action |
|---|---|---|---|---|---|---|
| 1 | UK1UK1EK2 | IK2 | Santun dalam kendaraan | apakah dalam menjalankan kendaraanya santun | - E-Learning<br>-<br>Penumpang | |
| 2 | UK1UK1EK3 | IK3 | Tidak menimbulkan dampak yang membahayakan kepada penumpang dan pengguna jalan lainya | Apakah membahayakan penumpang dan pengguna Jalan lainnya | - Pelatih<br>-<br>Penumpang | |

## Figure 5 Graduation Indicators

5. Assessment System
The assessment system is taken from the graduation indicator data by giving questions to e-learning, supervisors, trainers, passengers as in Figure 6.



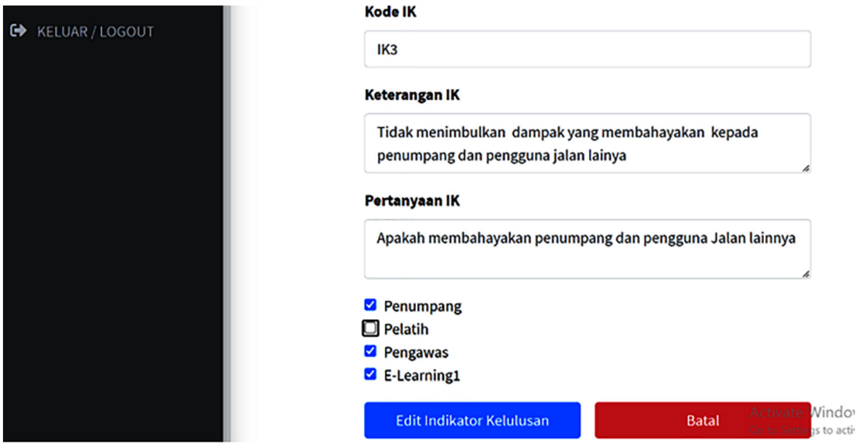
Sistem Pengelolaan Pengemudi Transportasi Umum Nasional
Berbasis Standar Kelulusan Kompetensi ( SKL)

Data Indikator Kelulusan

| # | KODE EK | KODE IK | Indikator Kelulusan (IK) | Pertanyaan | Pertanyaan Untuk | Action |
|---|---|---|---|---|---|---|
| 1 | UK1UK1EK3 | IK3 | Tidak menimbulkan dampak yang membahayakan kepada penumpang dan pengguna jalan lainya | Apakah membahayakan penumpang dan pengguna Jalan lainnya | - E-Learning<br>- Pengawas<br>- Pelatih<br>- Penumpang | |
| 2 | UK1UK1EK4 | IK4 | Mematuhi aturan dan ketentuan lalu lintas yang berlaku | Apakah pengemudi mengetahui ketentuan lalu lintas yang berlaku | - Pengawas<br>- Penumpang | |

**Figure 6 Assessment System**

6. Questions
Questions are taken from the graduation indicator data by checking the bottom button for passengers, trainers, supervisors, e-learning as in figure 7.



**Kode IK**
IK3

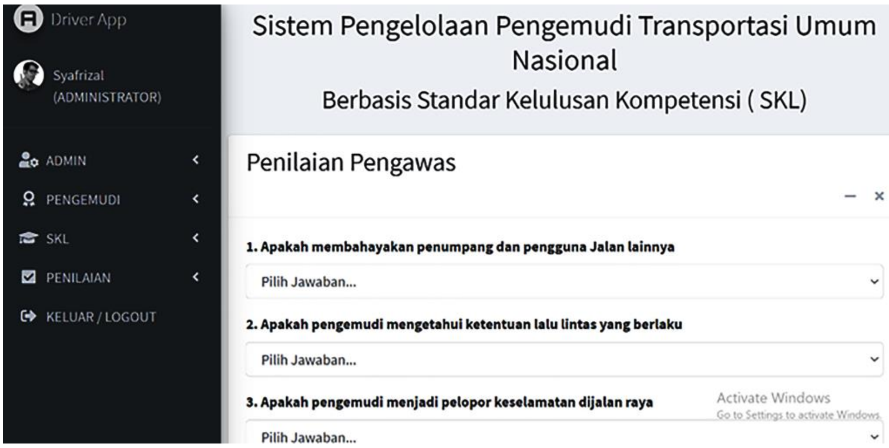
**Keterangan IK**
Tidak menimbulkan dampak yang membahayakan kepada penumpang dan pengguna jalan lainya

**Pertanyaan IK**
Apakah membahayakan penumpang dan pengguna Jalan lainnya

☑ Penumpang
☐ Pelatih
☑ Pengawas
☑ E-Learning1

**Gambar 7 Sistem Pertanyaan**

7. Questions for Supervisors
Questions are obtained from the question system in Figure 4.20, if the supervisor is checked, then the questions displayed to assess the driver are carried out by the supervisor as in Figure 8.



Sistem Pengelolaan Pengemudi Transportasi Umum Nasional
Berbasis Standar Kelulusan Kompetensi ( SKL)

Penilaian Pengawas

**1. Apakah membahayakan penumpang dan pengguna Jalan lainnya**
Pilih Jawaban...

**2. Apakah pengemudi mengetahui ketentuan lalu lintas yang berlaku**
Pilih Jawaban...

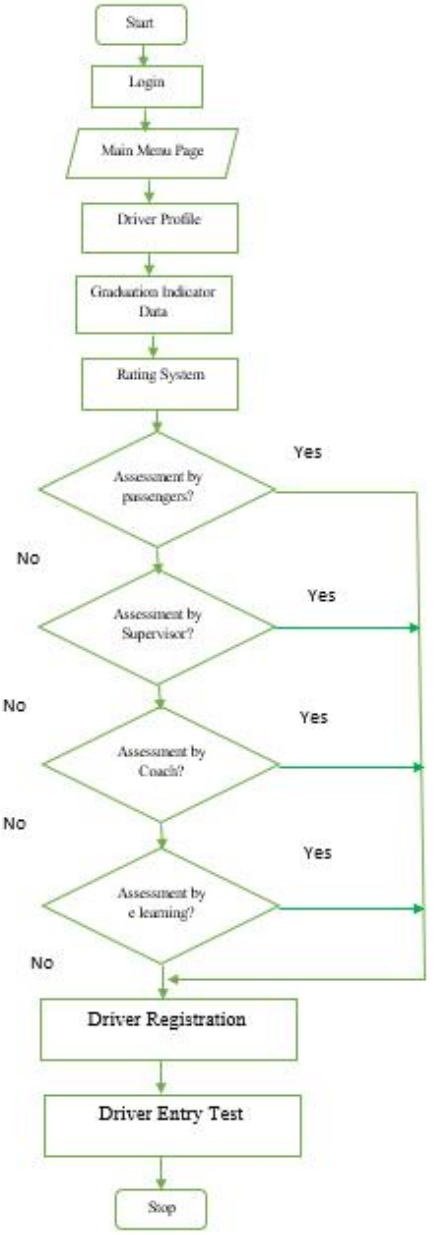
**3. Apakah pengemudi menjadi pelopor keselamatan dijalan raya**
Pilih Jawaban...

**Figure 8 Questions for Supervisors**

Testing Stages consist of, Flowchart/Sourcode Analysis, create Flow Graph, Cyclomatic Complexity, Independent Path Determination, Test Case Design as illustrated in Figure 2.

Testing Stages consist of, Flowchart/Sourcode Analysis, create Flow Graph, Cyclomatic Complexity, Independent Path Determination, Test Case Design

1. Flowchart Analysis
The first stage is the flowchart analysis of the White Box testing system with Path testing as illustrated in Figure 9.



Gambar 9 Analisa flowchat

2. Stage 2 Create a Flowgraph

The second stage is to create a flowchart of the White Box testing system with Path testing as depicted in Figure 10

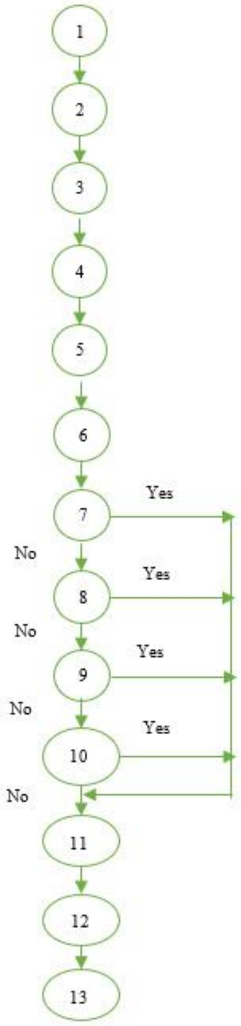


Figure 10 Flowgraph

3. Cylomatic Complexity

The third stage is the calculation of Cyclomatic Complexity of the White Box testing system with Path testing as below

V(G)= (E-N) +2
V(G)= Number of Regions
E= Number of edges determined by the arrow image
N= Number of graph nodes (nodes) with circle images
VG = (16- 13) + 2
VG = 5

4. Independent path

The fourth stage is determining the independent path of the White Box testing system with Path testing as below

Path 1 = 1,2,3,4,5,6,7,11,12.13
Path 2 = 1,2,3,4,5,6,8,11,12,13
Path 3 = 1,2,3,4,5,6,9,11,12,13
Path 4 = 1,2,3,4,5,6,10,11,12,13

### 5. Test Case Design

The fifth stage is the design of test cases taken from the independent paths that have been created. as follows

| | |
|---|---|
| Path | 1 |
| Track | 1,2,3,4,5,6,7,11,12.13 |
| Scenario | 1. Start<br>2. Login<br>2. Main Menu Page<br>3. Driver Profile<br>4. Graduation Indicator Data<br>5. Rating System<br>6. Assessment by passengers?<br>If Yes<br>11.Driver Registration<br>12. Driver Entry Test<br>13.Stop |
| Test Results | **Succeed** |
| Path | 2 |
| Track | 1,2,3,4,5,6,8,11,12,13 |
| Scenario | 1. Start<br>**2. Login**<br>3. Main Menu Page<br>4. Driver Profile<br>5. Graduation Indicator Data<br>6. Rating System<br>7. Assessment by Supervisor?<br>**If Yes**<br>11. Driver Registration<br>12.Driver Entry Test<br>13.Stop |
| Test Results | **Succeed** |
| Path | 3 |

| | |
|---|---|
| Track | **1,2,3,4,5,6,9,11,12,13** |
| Scenario | 1. Start<br>2. Login<br>3. Main Menu Page<br>4. Driver Profile<br>5. Graduation Indicator Data<br>6. Rating System<br>10. Assessment by Coach?<br>**If Yes**<br>11. Driver Registration<br>12.Driver Entry Test<br>13.Stop |
| Test Results | **Succeed** |
| Path | 4 |
| Jalur | 1,2,3,4,5,6,10,11,12,13 |
| Scenario | 1. Start<br>**2. Login**<br>3. Main Menu Page<br>4. Driver Profile<br>5. Graduation Indicator Data<br>6. Rating System<br>8. Assessment by e learning??<br>**If Yes**<br>14. Driver Registration<br>15.Driver Entry Test<br>16.Stop |
| Test Results | **Succeed** |

### Conclusions

The application of Path Testing as one of the techniques in White Box Testing on the national public transportation driver behavior and competency assessment system has proven effective in identifying program execution paths that contain logical errors. By evaluating all independent paths in the program code, this test is able to ensure that each control flow has been thoroughly tested. The results of this test indicate that the system works according to the expected logic, and is able to provide accurate output based on the input given.

### Suggestion

The application of Path Testing in White Box Testing for the National Public Transportation Driver Behavior and Competency Assessment System is recommended that each logical path in the program code be documented systematically. This makes the path testing process easier in the future. path testing is integrated with automated testing tools so that the testing process can be carried out repeatedly and consistently every time there is a change in the code.

## Reference

[1] M. Kumar, A. Professor, S. Kumar Singh, R. K. Dwivedi, and A. Professor, "International Journal of Advance Research in Computer Science and Management Studies," International Journal of Advance Research in Computer Science and Management Studies, vol. 3, no. 10, 2015, [Online]. Available: www.ijarcsms.com

[2] H. Gusdevi et al., "PENGUJIAN WHITE-BOX PADA APLIKASI DEBT MANAGER BERBASIS ANDROID," vol. 04, 2022.

[3] A. Dimas Saputro and A. Azzahra Narwastika, "IMPLEMENTASI WHITE BOX TESTING DENGAN TEKNIK BASIS PATH PADA PENGUJIAN FORM PEMINJAMAN SISTEM APLIKASI PERPUSTAKAAN."

[4] M. A. Nurwicaksono, I. N. Lisa, A. R. Tiara, and R. Sidik, "Optimasi Sistem Informasi Konsultasi Hukum melalui Pendekatan Pengujian Kombinasi White-box dan Black-box," Jurnal Manajemen Informatika (JAMIKA), vol. 14, no. 1, pp. 1–15, Nov. 2023, doi: 10.34010/jamika.v14i1.10110.

[5] S. Y. Rini and A. Kusmaya Putri, "IMPLEMENTASI BLACKBOX TESTING DAN WHITEBOX TESTING PADA PENGUJIAN FORM PROFIL TOKO ADMIN SISTEM APLIKASI RAJA ONGKIR BERBASIS WEBSITE."

[6] Enrique. Peñalosa, Fernando. Votero, Paloma. Balencia, and Roi. Barreras, Licuadoras luminosas. La Jaula Publicasiones, 2019.

[7] M. M. Syaikhuddin, C. Anam, A. R. Rinaldi, and M. Conoras, "Conventional Software Testing Using White Box Method. Kinetik : Game Technology, Information System," Computer Network, Computing, Electronics, and Control, vol. 3, no. 1, pp. 67–74, 2018.

[8] G. Hoendarto and S. Tendean, "PENGGUNAAN METODE KOTAK HITAM DAN KOTAK PUTIH DALAM MENGUJI SEBUAH PRODUK SISTEM INORMASI."

[9] Rahman Abdillah, Rudi Hermawan, Wawan Hermawansyah, Ibnu Adkha, and Heri Arifin, "Pengujian Perangkat Lunak Sistem Informasi Inventori pada Usaha Jasa Pengiriman Paket," Polygon : Jurnal Ilmu Komputer dan Ilmu Pengetahuan Alam, vol. 2, no. 4, pp. 166–175, Jul. 2024, doi: 10.62383/polygon.v2i4.199.

[10] D. Hamlet, "Software Quality, Software Process, and Software Testing," 1995.

[11] Safrizal.et al, "TESTING_DAN_IMPLEMENTASI". PT Mafy Media Literasi Indonesia,ISBN 978-623-8638-29-1

[12] N. Li, U. Praphamontripong, and J. Offutt, "An experimental comparison of four unit test criteria: Mutation, edge-pair, all-uses and prime path coverage," in IEEE International Conference on Software Testing, Verification, and Validation Workshops, ICSTW 2009, 2009, pp. 220–229. doi: 10.1109/ICSTW.2009.30.

[13] N. Golian, V. Golian, and I. Afanasieva, "BLACK AND WHITE-BOX UNIT TESTING FOR WEB APPLICATIONS," Bulletin of National Technical University "KhPI". Series: System Analysis, Control and Information Technologies, no. 1 (7), pp. 79–83, Jul. 2022, doi: 10.20998/2079-0023.2022.01.13.

[14] S. Izzat and N. N. Saleem, "Software Testing Techniques and Tools: A Review," Journal of Education and Science, vol. 32, no. 2, pp. 31–40, Jun. 2023, doi: 10.33899/edusj.2023.137480.1305.

[15] R. Andrian Ibrahim and G. Saktian Laksito, "Optimization of White Box Testing by Utilizing Branching and Repeating Structures in Java Programs Using Base Path," International Journal of Mathematics, Statistics, and Computing, vol. 2, no. 2, pp. 85–89, 2024.

[16] A. Sekhon, Y. Ji, M. B. Dwyer, and Y. Qi, "White-box Testing of NLP models with Mask Neuron Coverage," May 2022, [Online]. Available: http://arxiv.org/abs/2205.05050

[17] P. M. Chawan, W. : Www, S. Thakare, S. Chavan, and M. Chawan, "Software Testing Strategies and Techniques International Journal of Emerging Technology and Advanced Engineering Software Testing Strategies and Techniques," 2012. [Online]. Available: www.ijetae.com

[18] B. Žlahtič, J. Završnik, H. Blažun Vošner, and P. Kokol, "Transferring Black-Box Decision Making to a White-Box Model," Electronics (Switzerland), vol. 13, no. 10, May 2024, doi: 10.3390/electronics13101895.

[19] T. Sheakh, T. Hussain, and S. Singh, "International Journal of Allied Practice, Research and Review A Comparative Study of Software Testing Techniques Viz. White Box Testing Black Box Testing and Grey Box Testing," IJAPRR International Peer Reviewed Refereed Journal, vol. II, pp. 1–08, 2015, [Online]. Available: https://www.researchgate.net/publication/276028491

[20] S. Bardin, O. Chebaro, M. Delahaye, and N. Kosmatov, "An all-in-one toolkit for automated white-box testing," in Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), Springer Verlag, 2014, pp. 53–60. doi: 10.1007/978-3-319-09099-3_4.

[21] F. Lammermann and S. Wappler, "Benefits of Software Measures for Evolutionary White-Box Testing." [Online]. Available: https://www.researchgate.net/publication/327691019

[22] A. Caniço and A. Santos, "Witter: A Library for White-Box Testing of Introductory Programming Algorithms," in SPLASH-E 2023 - Proceedings of the 2023 ACM SIGPLAN International Symposium on SPLASH-E, Co-located with: SPLASH 2023, Association for Computing Machinery, Inc, Oct. 2023, pp. 69–74. doi: 10.1145/3622780.3623650.

[23] A. Suryanta, L. Wahyu Widianti, dan Mirsya Ashari, P. Kodiklatad, and S. Jakarta STI, "PENGUJIAN WHITE BOX TERHADAP SISTEM INFORMASI AKADEMIK (SIAKAD) DI POLITEKNIK ANGKATAN DARAT," 2025.

[24] G. W. Sasmito, "White Box Testing with Basis Path Technique in the Demography Administration Website," in ICSECC 2020 - 2nd International Conference on Sustainable Engineering and Creative Computing, Proceedings, Institute of Electrical and Electronics Engineers Inc., 2020, pp. 86–92. doi: 10.1109/ICSECC51444.2020.9557428.

[25] A. H. Ali and N. N. Saleem, "Data Flow Testing and Tools Review," Journal of Education and Science, vol. 32, no. 2, pp. 51–60, Jun. 2023, doi: 10.33899/edusj.2023.137611.1315.