ATCSJ
Applied Technology and Computing Science Journal

Open Access **p-ISSN 2621-4458 | e-ISSN 2621-4474**

Available online to journal2.unusa.ac.id

UNUSA

S4-Accredited – SK No.79/M/KPT/2023

Journal Page is available to https://journal2.unusa.ac.id/index.php/ATCSJ

Integration of ITIL V3, COBIT 5, and Service Desk Standards in IT Service Desk Design

Afni Virda Alfadina[1], Endang Sulistiyani[2,*],Rizqi Putri Nourma Budiarti[3]

[1,2,3] *Information System, Universitas Nahdlatul Ulama Surabaya, Indonesia*
Jl. Raya Jemursari 51-57, Surabaya, Indonesia
[1]afnivirda047.if20@student.unusa.ac.id, [2*]sulistiyani.endang@unusa.ac.id ,[3]rizqi.putri.nb@unusa.ac.id

Article history:

Received 20 October 2025
Revised 10 November 2025
Accepted 28 November 2025
Available online 2 December 2025

Keywords:

Software Testing,
Test Case Generation,
UML Diagrams,
Depth-First Search (DFS) Algorithm,
Automated Testing.

Abstract

Software testing is a critical phase in the software development life cycle, consuming more than 50% of the total time and resources. One important activity within this phase is the creation of test cases, which can be problematic, particularly when done manually, due to the significant time and resources required. To address this, test case generation can be automated using two main approaches: based on program code or design models. Generating test cases at the design level using UML has been shown to be more time- and cost-efficient than generating them from code. Previous studies have focused on creating test cases using a single UML diagram or a combination of two UML diagrams, producing satisfactory results but with limitations in covering all possible execution paths of the software. To overcome these limitations, this study proposes a combination of three UML diagrams such as activity diagrams, sequence diagrams, and state diagrams using the Depth-First Search (DFS) algorithm. The proposed approach involves three major stages: preparation, execution, and evaluation. By combining these three UML diagrams, it is expected that the method will generate the maximum number of test cases, ensuring comprehensive coverage of all software paths with minimal issues.

I. INTRODUCTION

Information Technology Service Management (ITSM) is a set of procedures implemented by an organization to improve the operational quality of IT systems [1]. ITSM refers to the implementation and management of quality IT services that meet organizational needs, delivered by IT service providers through a combination of people, processes, and technology [2]. One key element in ITSM implementation is the service desk, which functions as a Single Point of Contact (SPOC) for all IT service users. It is not

only responsible for addressing communication needs of users and IT staff but also for achieving goals from both customer and service provider perspectives [3]. The service desk plays a crucial role in handling issues such as receiving incident reports, recording incidents, providing solutions to problems, and escalating incidents to second-line or third-line support if the service desk cannot resolve them [4].

Nahdlatul Ulama University Surabaya (UNUSA) has a dedicated unit, the Directorate of Information Systems, responsible for program, network, and hardware development [5]. However, UNUSA faces challenges in its services due to the lack of a formal IT Service Desk. While help desk technology exists, there is no centralized framework or best practice to manage and update service desk knowledge. In some organizations, this may result in problem resolution being entirely dependent on personal experience without a structured methodology [3]. Incidents outside the scope of the support hotline also pose potential issues [6]. Implementing a framework or best practice in service desk development can assist organizations in IT development and operations, improving end-user support and customer satisfaction. It also enables proactive problem identification and effective communication with each end user [3].

A 2017 survey of 261 IT leaders in large organizations worldwide revealed that the most widely used frameworks or best practices were ITIL at 47% and COBIT at 36% [7]. Despite widespread adoption of ITIL V3 and COBIT in service desk implementation, ITIL V3 still has limitations, including framework complexity [4]. ITIL V3 also lacks its own assessment standard. In service desk organizational structure development, defining each function and responsibility is required. Nevertheless, ITIL V3 empowers all IT service stakeholders in management [8]. COBIT, on the other hand, is primarily used to assess the maturity level of IT processes within an organization. While COBIT is generally considered a self-assessment tool for IT management, it does not specifically focus on the maturity level of service desk-provided services [9].

Meanwhile, the Service Desk Standard focuses on maturity levels and can be used to assess the performance of service desk management processes and role competencies [10]. COBIT 5 provides explicit assignment and accountability of stakeholders using the Responsible, Accountable, Consulted, and Informed (RACI) mapping [11], optimizing role definition within the service desk. By mapping actor roles according to COBIT 5 alongside the organizational structure, recommendations for procedures, policies, and solutions can be generated [12].

Based on the identified limitations in ITIL V3 and COBIT 5 and the potential of the Service Desk Standard, this study aims to design an IT Service Desk integrating ITIL V3, COBIT 5, and Service Desk Standard. Although ITIL has a newer version, ITIL V4, offering 34 comprehensive practices for service management [13], it does not provide detailed guidance for the organizational structure of the service desk. ITIL V4 primarily addresses new service management challenges using modern technology [14] without detailing service organizational structure [15]. Therefore, this approach is expected to overcome the weaknesses of the previous frameworks and optimize the use of human resources and processes to improve IT service quality at UNUSA.

 E-ISSN:*2621-4474, ISSN:2621-4458.*

II. LITERATURE REVIEW

In this section, the researcher presents references used in the study, including previous research that is similar to the current study. The first study, titled "Analysis of Service Desk Structure in Higher Education (Case Study: Sepuluh Nopember Institute of Technology Surabaya)", produced recommendations for a service desk structure suitable for the university. The difference in this study lies in the integration of ITIL V3 guidance, which provides directions related to IT service desk structure, and the COBIT 5 framework, which can be used to map a RACI Chart to define roles and responsibilities tailored to the specific conditions of the higher education institution under study (ATGSD) [7].

The second study, titled "A Study of Service Desk Setup in Implementing IT Service Management in Enterprises", found that key performance indicators (KPIs) of the service desk improved after implementing ITIL best practices, considering three perspectives: process, people, and technology. The difference in this study is the use of a combination of ITIL V3 and COBIT 5 frameworks to define roles and responsibilities within the IT service desk structure, as well as the integration of ITIL V3, COBIT 5, and Service Desk Standards for mapping IT processes [5].

The third study, titled "Advantages of the Implementation of Service Desk Based on ITIL Framework in the Telecommunication Industry", found that using ITIL in service desk implementation led to better outcomes across seven service desk processes, excluding service catalog management and access management. The difference in this study lies in combining ITIL V3, COBIT 5, and Service Desk Standards to map each process, such as request fulfillment, incident, access, and problem management. Additionally, the Service Desk Standard was used to measure maturity levels [12].

The fourth study, titled "Evaluation of IT Service Desk: A Case Study", resulted in recommendations for a new service desk process structure and clarified stakeholder roles. The main difference in this study is the design of the IT service desk structure involving the definition of roles and responsibilities. This approach uses a combination of ITIL V3 and COBIT 5 frameworks and applies a RACI Chart as a supporting tool [13].

The fifth study, titled "ITSM Analysis Using ITIL V3 in Service Operation at PT. Inovasi Tjaraka Buana", produced workflow designs for incident management and problem management aligned with ITIL V3 processes in operational services. The difference in this study lies in mapping the five core processes implemented in the service desk, referencing ITIL V3, COBIT 5, and Service Desk Standards [14].

The sixth study, titled "Incident Management of Information Technology in Indonesian Higher Education Based on COBIT Framework: A Review", found that the maturity level of IT management was still very low (around level 2) and indicated the absence of technical guidelines for IT incident management procedures. The difference in this study is the integration of ITIL V3, COBIT 5, and Service Desk Standards to map processes such as request fulfillment, incident, access, and problem management, and the use of Service Desk Standards to measure maturity [6].

The seventh study, titled "Design of Hospital Management Information System (SIMRS) Governance Based on ITIL V3 & Service Desk Standard (Case Study: RSUD Kota Kediri)", resulted in an IT governance design for SIMRS with one IT organizational structure, 11 IT processes, and documentation consisting of six guidelines, five SOPs, and ten templates or forms. The difference in this study is the design of the IT service desk structure combining ITIL V3, COBIT 5, and Service Desk Standards for mapping service desk processes [15].

The eighth study, titled "Evaluating the Quality of a Help Desk Complaint Management Service Using Six-Sigma and COBIT 5 Framework", found improvements in the DMAIC sub-methodology by adopting COBIT 5 in the DSS02 (Manage Service) and DSS03 (Manage Problems) domains, resulting in 12 recommendations for policy procedures and solution procedures. The difference in this study is the actor mapping process within the service desk structure using a combination of ITIL V3 and COBIT 5 frameworks from DSS domains with a RACI Chart [3].

The ninth study, titled "Development of Risk-Based Audit Tools Based on COBIT 5 and Service Desk Standards for Service Desk (Case Study: Directorate of Technology and Information Systems Development, ITS)", found that main operational risks in the service desk include misinterpreting user requests, delayed responses, and user dissatisfaction. The control objective to mitigate these risks involves defining services, fulfilling requests, and handling incidents. The difference in this study is the design of the service desk structure using process mapping with ITIL V3, COBIT 5 DSS domains, and Service Desk Standards [8].

The tenth study, titled "The Measurement of Maturity Level of Information Technology Service Based on COBIT 5 Framework", found that the current IT service maturity level is at level 3 (defined) on a scale of 5 (optimized), indicating that implemented and documented processes follow the established standards and produce positive results. The difference in this study is the focus on designing the service desk structure using best practices from Service Desk Standards to measure maturity specifically within the service desk [9].

III. METHODS

This section outlines the methodological approach applied in this study. The approach is crucial to provide a systematic guideline throughout the research process. Each methodology step is explained in detail, as shown in Figure 1.

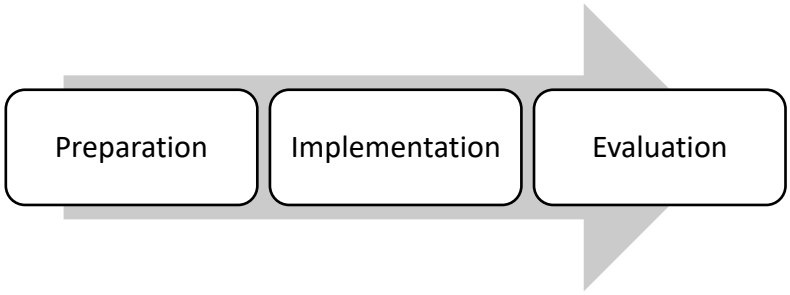

Figure 1. Research Methodology

Based on the research methodology presented in Figure 1, each research stage is described as follows:

A. Preparation Stage

The preparation stage is the initial phase of this study. This stage consists of several sub-steps:

1. Preparation of Instruments for Gathering Existing and Ideal Condition Requirements
   In this step, the researcher develops an interview protocol to gather data on the current condition. The protocol contains a list of questions to obtain information regarding the structure, roles, and processes of the IT service desk in the existing condition.
2. Measurement of Existing Condition Maturity Level
   In this step, the researcher evaluates the maturity level of the existing condition regarding the structure, roles, and processes, based on data obtained from the previous step.

B. Implementation Stage

1. Design of Proposed IT Service Desk Structure
   The researcher designs a recommended service desk structure based on the analysis of the existing and ideal conditions and the maturity assessment conducted in the preparation stage.
2. Definition of Proposed IT Service Desk Roles
   The researcher defines the roles of the IT Service Desk based on the previously designed structure.
3. Preparation of Standard Operating Procedures (SOP) for IT Service Desk
   The researcher develops SOP documents aligned with the Service Desk process mapping according to the ITIL V3 framework. The development follows the guidelines set by the Indonesian Ministry of Administrative Reform and Bureaucratic Reform Regulation No. 35 of 2012 (Permen PAN & RB No. 35/2012).
4. Formulation of Responsibility, Accountable, Consulted, and Informed (RACI) Chart

   In this step, the researcher formulates a RACI chart based on the proposed roles and processes outlined in the SOP for the IT Service Desk.

C. Evaluation Stage

The evaluation stage is the final phase of this study, consisting of the following sub-steps:

1. Assessment of Maturity Level of Proposed IT Service Desk
   The researcher reassesses the maturity level of the proposed IT Service Desk generated in previous stages.
2. Feedback Collection on Proposed IT Service Desk Design
   The researcher collects feedback on the proposed IT Service Desk design from relevant stakeholders.

IV. RESULTS AND DISCUSSIONS

This section explains the design process and the results obtained from the study, which serve as a guideline for the final project development.

A. Instruments for Gathering Existing and Ideal Conditions

To study the design of the service desk structure and determine the ideal condition, a mapping was conducted consisting of structure, roles, and processes.

Table 1. Instruments for Gathering Existing Conditions

Scope	**ITIL V3**	**COBIT 5**	**Service Desk Standard**
Service Desk Structure	• Local Service Desk • Centralized Service Desk • Virtual Service Desk • Follow The Sun • Spesialized Service Desk Group	**Not Available**	**Not Available**
Service Desk Roles	• Service Desk Manager • Service Desk Supervisor • Service Desk Analyst • Technical Support	Responsible, Accountable, Consulted, dan Informed (RACI) Chart	• Service Desk Manager • Service Desk Analyst
Service Desk Processes	• Event Management • Incident Management • Request Fulfilment • Problem Management • Access Management	• DSS01 Manage Operations • DSS02 Manage Service Requests and Incidents • DSS03 Manage Problems • DSS05 Manage Security Services	**Not Available**

From Table 1, it can be seen that ITIL V3, COBIT 5, and Service Desk Standards each play a role in designing the service desk structure, including structure, roles, and processes.

B. Ideal and Existing Conditions of the Service Desk

This section presents the ideal service desk condition based on structure, roles, and processes. The ideal condition is derived from ITIL V3, COBIT 5, and Service Desk Standards.

Table 2. Ideal and Existing Conditions

Procedure	**Description**
SOP for Email Request	Submitting a request to create a student email account with an attached letter

SOP for Zoom Meeting Request	Borrowing Zoom with 1000-user capacity (300 users should contact the program or faculty admin)
SOP for Smartclass Usage	Notification before using Smartclass
SOP for Laboratory	Usage of laboratories
SOP for Information System Services	Complaints regarding the information system used
SOP for SIM Development	Development of the information system in use
SOP for SIM Account Complaints	Handling issues such as forgotten passwords
SOP for Hardware Maintenance	Repair of computer hardware at Unusa
SOP for Wireless Connection Handling	Handling internet connection issues using wired connections

C. Standard Operating Procedure (SOP) for Request Fulfillment

The service categories and SOPs for request fulfilment are presented in Table 3**.**

Table 3. SOP Request Fulfillment

Service Category	Service Details	Service Provided	Responsible Party
Email	• UNUSA Email	Create new email accountChange passwordEmail errors	IT Service Desk Team
Software Programs	• Central SIM• Mobile Applications	Non-functional features in SIMDatabase additionRole/access changesNew system developmentFeature addition to existing system	Software Programming Technicians
Website	• Website• Domain• Hosting	Domain additionDomain repairWebsite template requestsNew domain/hosting requests	Website Technicians
Hardware	• Hardware (PC, printer, scanner, router, HDMI)	Provision of desktop and peripherals (mouse, keyboard, monitor)Hardware maintenance and repair	Hardware Technicians
Infrastructure and Network	• Internet connection• Server• Wi-Fi• VPN network	Internet access provisionNetwork troubleshootingServer	Infrastructure and Network Technicians

		downServer access issues	

This SOP (Standard Operating Procedure) table outlines the different categories of IT-related service requests within an organization, specifying the services provided and the responsible parties for each. Email requests are handled by the IT Service Desk, covering account creation, password management, and email errors. Software-related issues, including database updates and system development, are managed by Software Programming Technicians. Website and domain-related services are handled by Website Technicians, while Hardware Technicians take care of desktops, peripherals, and maintenance tasks. Finally, Infrastructure and Network Technicians are responsible for internet connectivity, server management, Wi-Fi, and VPN network issues. This structured approach ensures that each request type is efficiently addressed by the appropriate technical team.

V. CONCLUSIONS

Based on the analysis of test case results from each diagram, the conclusions of this final project research are as follows: The test cases generated from the activity diagram, sequence diagram, state diagram, combinations of activity and sequence diagrams, activity and state diagrams, sequence and state diagrams, and the combination of activity, sequence, and state diagrams all produced the same number of test cases for each system feature. The system has nine features: login with 2 test cases, new account registration with 2, certificate checking with 2, registration with 2, grades with 1, certificate issuance with 1, profile with 1, password change with 2, and logout with 1. Test cases with 2 represent both successful and failed scenarios, while those with 1 represent only successful scenarios. The combination of the three UML diagrams such as activity, sequence, and state. It's performed best in terms of path coverage, generating the largest number of test paths and providing the most extensive and in-depth coverage. This combination captures multiple system aspects from different perspectives, offering a more comprehensive understanding of possible system paths. In terms of data flow coverage, both individual and combined diagrams performed well, ensuring all data flows were included without duplication. Therefore, using a combination of various diagrams in test case creation is crucial for achieving optimal path and data flow coverage. Based on these results and conclusions, suggestions for future research include developing methods or tools to automate the creation of system testing graphs, which are currently done manually and require significant time, and exploring other combinations of three diagrams beyond activity, sequence, and state diagrams to evaluate their performance.

VI. REFERENCES

[1] M. Khandai, A. A. Acharya, and D. P. Mohapatra, "Test case generation for concurrent system using UML combinational diagram," *Int. J. Comput. Sci. Inf. Technol. IJCSIT*, vol. 2, no. 3, pp. 1172–1181, 2011.

[2] A. V.K.Shanthi and G. MohanKumar, "A Novel Approach for Automated Test Path Generation using TABU Search Algorithm," *Int. J. Comput. Appl.*, vol. 48, no. 13, pp. 28–34, 2012, doi: 10.5120/7410-0449.

[3] Meiliana, I. Septian, R. S. Alianto, Daniel, and F. L. Gaol, "Automated Test Case Generation from UML Activity Diagram and Sequence Diagram using Depth First Search Algorithm," *Procedia Comput. Sci.*, vol. 116, pp. 629–637, 2017, doi: 10.1016/j.procs.2017.10.029.

[4] W. Li, F. Le Gall, and N. Spaseski, "A survey on model-based testing tools for test case generation," *Commun. Comput. Inf. Sci.*, vol. 779, pp. 77–89, 2018, doi: 10.1007/978-3-319-71734-0_7.

[5] A. K. Jena, S. K. Swain, and D. P. Mohapatra, "A novel approach for test case generation from UML activity diagram," *Proc. 2014 Int. Conf. Issues Challenges Intell. Comput. Tech. ICICT 2014*, pp. 621–629, 2014, doi: 10.1109/ICICICT.2014.6781352.

[6] D. Olianas, "Automatic test case generation from UML State machine diagrams," *ACM Int. Conf. Proceeding Ser.*, vol. 42, no. 7, pp. 26–36, 2019, doi: 10.1145/3328433.3328465.

[7] V. Panthi and D. P. Mohapatra, "Automatic test case generation using sequence diagram," *Adv. Intell. Syst. Comput.*, vol. 174 AISC, no. 4, pp. 277–284, 2013, doi: 10.1007/978-81-322-0740-5_33.

[8] S. K. Swain, D. P. Mohapatra, and R. Mall, "Journal of Object Technology Test Case Generation Based on State and Activity Models," vol. 9, no. 5, pp. 1–27, 2010, [Online]. Available: http://www.jot.fm/issues/issue_2010_09/article1.pdf

[9] N. Khurana and R. S. Chillar, "Test Case Generation and Optimization using UML Models and Genetic Algorithm," *Procedia Comput. Sci.*, vol. 57, pp. 996–1004, 2015, doi: 10.1016/j.procs.2015.07.502.

[10] K. Jin and K. Lano, "Generation of Test Cases from UML Diagrams - A Systematic Literature Review," *ACM Int. Conf. Proceeding Ser.*, 2021, doi: 10.1145/3452383.3452408.

[11] A. Tripathy and A. Mitra, "Test case generation using activity diagram and sequence diagram," Adv. Intell. Syst. Comput., vol. 174 AISC, pp. 121–129, 2013, doi: 10.1007/978-81-322-0740-5_16.

[12] R. K. Swain, "Minimal Testcase Generation for Object-Oriented Software with State Charts," Int. J. Softw. Eng. Appl., vol. 3, no. 4, pp. 39–59, 2012, doi: 10.5121/ijsea.2012.3404.

[13] S. Lorpunmanee, S. Kamonsantiroj, and L. Pipanmaekaporn, "A memorization approach for test case generation in concurrent UML activity diagram," ACM Int. Conf. Proceeding Ser., vol. Part F1482, pp. 20– 25, 2019, doi: 10.1145/3318236.3318256.

[14] S. S. Panigrahi, S. Shaurya, P. Das, A. K. Swain, and A. K. Jena, "Test Scenarios Generation Using UML Sequence Diagram," Proc. - 2018 Int. Conf. Inf. Technol. ICIT 2018, no. I, pp. 50–56, 2018, doi: 10.1109/ICIT.2018.00022.

[15] A. Arwan and D. Sagita, "Automation Of Independent Path Searching using Depth First Search," J. Inf. Technol. Comput. Sci., vol. 3, no. 1, pp. 104–112, 2018, doi: 10.25126/jitecs.20183162.