JURNAL TEKNOLOGI DAN OPEN SOURCE
Vol. 8, No. 1, June 2025, pp. 208~216
e-ISSN: 2622-1659, accredited Four Grade by Kemenristekdikti, Decree No: 152/E/KPT/2023
DOI: 10.36378/jtos.v8i1.4413

Testing of Thesis Management Application Using STLC Framework and Automation Tools in Physics Study Program UPN "Veteran" Jatim

Dewangga Nanda Arjuna[1], Mohamad Irwan Afandi[2], Abdul Rezha Efrat Najaf[3]
[1,2,3]Department of Information System, Faculty of Computer Science, UPN "Veteran" Jawa Timur, Surabaya

Article Info

Article history:

Received 05 15, 2025
Revised 05 20, 2025
Accepted 06 11, 2025

Keywords:

Management,
Scrum,
STLC,
Testing
Thesis,

ABSTRACT

Thesis administration at the higher education level, such as in the Physics Study Program, still faces challenges due to the absence of an integrated system. To address this issue, a web-based Thesis Management Application was developed to support five main user roles: Students, Lecturers, Thesis Coordinators, Program Coordinators, and Administrators. The system offers a comprehensive set of features including authentication, pre-proposal submission, advisor scheduling, seminar and exam management, publication submission, oral examinations, and final graduation requirements. This study focuses on the Quality Assurance (QA) process using the Scrum methodology and Software Testing Life Cycle (STLC) framework over six sprints conducted within three months. The process began with the mapping of 225 functional requirements on the Front-End and 206 on the Back-End, which were then translated into 1240 test cases derived from various usage scenarios, authorization validations, and interaction flow explorations. Non-functional testing was also conducted at the end of the sixth sprint, covering aspects such as portability, usability, performance, availability, and security. The results show that all system requirements were successfully met, indicating that the application is feasible for supporting a structured thesis administration process.

Corresponding Author:

Dewangga Nanda Arjuna
Department of Information System
Faculty of Computer Science
UPN "Veteran" Jawa Timur
Surabaya
Email: adewanggan@gmail.com

1. Introduction

Technological advancements, particularly the internet, have significantly impacted the field of education, enabling faster and broader access to information and encouraging self-directed learning without time and location constraints [1]. The internet also serves as a vital tool for supporting student research activities, including reference searches and data collection for thesis writing [2], [3], [4], [5]. A thesis is a crucial component of a student's academic journey, reflecting critical thinking skills and the application of knowledge acquired during their studies [6]. The thesis process involves various stages, from title submission to the final defense, requiring coordination between students, lecturers, and academic staff [7], [8].

The Physics Study Program at Universitas Pembangunan Nasional "Veteran" Jawa Timur, established in 2022, identified the need for a thesis management system to support its lengthy and complex administrative processes. Case studies from the Information Systems and Informatics Study Programs show that the presence of a thesis management system can accelerate and simplify the thesis completion process [9], [10]. A web-based Thesis Management Application was designed to support the thesis process from the proposal seminar to the final seminar, involving five primary user roles: Student, Lecturer, Thesis Coordinator, Study Program Coordinator, and Study Program Admin. The application features seminar registration, automated scheduling, examiner mapping, assessment, and document revision management [9], [10]. The system is developed using a client-server approach, with the front-end serving the user interface and the back-end managing system logic and the database. Advantages of a web-based system include accessibility, flexibility, and efficient distribution [11], [12], [13]. System testing includes both functional and non-functional aspects such as security, performance, and availability to ensure application quality and user satisfaction [14], [15]. Testing plays a crucial role in detecting bugs early and ensuring the application meets user requirements [16], [17].

The Software Testing Life Cycle (STLC) methodology was implemented to test this application, as it provides a structured approach from requirements analysis to test result evaluation. STLC enhances team communication processes and ensures the reliability of each tested component [18], [19]. Automation tools such as Katalon Studio, Postman, and JMeter support the efficiency and consistency of the testing process, enabling repeated testing with accurate results and faster detection of software errors [20].This study aims to evaluate the quality of the web-based Thesis Management Application for the Physics Study Program at UPN "Veteran" Jawa Timur using the STLC approach and automation tools, ensuring the application can optimally and sustainably support thesis administrative needs.

2. Research Method

This chapter outlines the steps undertaken to achieve the research objectives. The research methodology, adapted from a previous study [21], includes data collection and a testing framework based on the Software Testing Life Cycle (STLC) within the SCRUM framework, as illustrated in Figure 1.

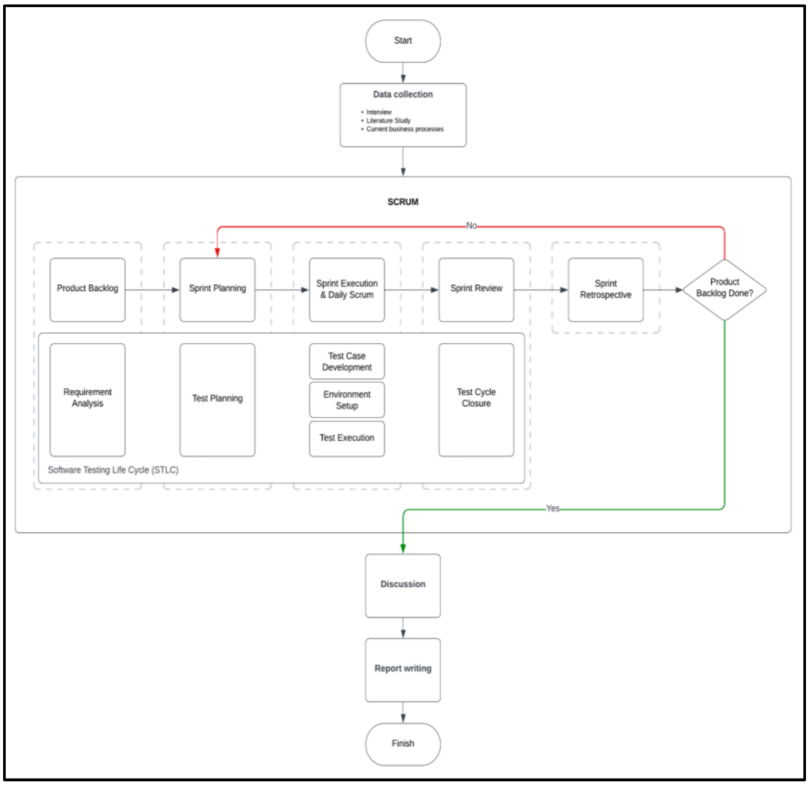

Figure 1. Research Design Diagram

These stages are described as follows:

2.1. Data Collection

Data were gathered through interviews and literature studies to obtain essential information for the research. The collected data served as a reference for analyzing system requirements and guiding the system design process.

2.2. Scrum dan STLC

Scrum is a framework designed to address the challenges of complex software development by applying Agile principles through structured activities such as sprint, sprint planning, daily scrum, sprint review, and sprint retrospective [22], [23].

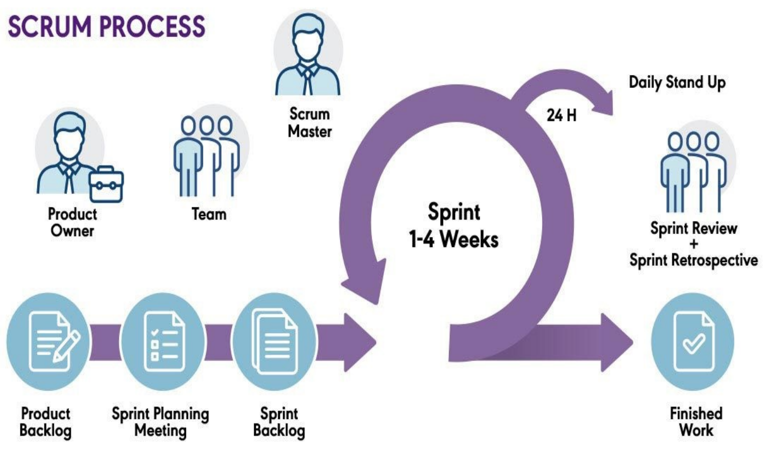

Figure 2. Scrum Methodology (Source: PM Partners)

The Software Testing Life Cycle (STLC) is a structured sequence of activities carried out to test software systematically. It involves well-planned stages executed in an organized manner to ensure optimal testing outcomes. The main phases of STLC include Requirement Analysis, Test Planning, Test Case Development, Environment Setup, Test Execution, and Test Cycle Closure [24]. In this study, Scrum was adopted as the development methodology, with the STLC process implemented concurrently alongside Scrum.

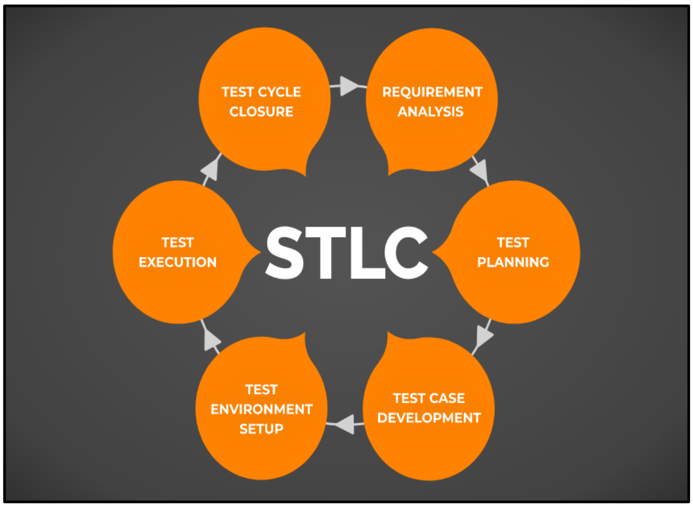

Figure 3. STLC Methodology (Source: RareCrew)

a) Product Backlog and Requirement Analysis

The Product Backlog stage organizes and prioritizes development tasks to keep the team focused on key goals. In the STLC process, after feature priorities are set, Requirement Analysis is conducted to clarify system requirements and ensure clear Quality Assurance understanding of the software under test [23], [25].

b) Sprint Planning and Test Planning

Tasks for the Sprint are selected based on priority. Test Planning defines testing strategies, resources, environments, and schedules to ensure clear testing direction [23], [25].

c) Sprint Execution dan Daily Scrum

The team works on assigned tasks while Quality Assurance prepares test cases, sets up the test environment, and executes functional and non-functional tests [23], [25].

d) Sprint Review dan Test Cycle Closure

Sprint outcomes are reviewed, and QA summarizes testing results, identifies improvements, and assesses software readiness for release [23], [25].

e) Sprint Retrospective

The team evaluates the Sprint process to identify obstacles, improve efficiency, strengthen collaboration, and define corrective actions for the next cycle [23], [25].

2.3. Discussion

This section reflects on the use of testing tools throughout the process, discussing their functionalities, usability, and limitations encountered during testing. The insights are intended to support future researchers interested in applying similar tools for functional and non-functional testing.

3. Result and Discussion

This chapter presents the outcomes of each stage of the research, including data collection, the implementation of Scrum and STLC methodologies, and the evaluation results of the Thesis Management Application based on predefined requirements.

3.1. Product Backlog and Requirement Analysis

The system development began by compiling a prioritized product backlog based on interviews with the Thesis Coordinator of the Physics Department at UPN “Veteran” Jawa Timur. This was followed by a requirement analysis phase where the QA team thoroughly reviewed both functional and non-functional needs to ensure comprehensive test coverage. The system was designed to support the entire thesis management workflow, involving user roles such as students, supervisors, coordinators, and administrators, with features including title submission, seminar scheduling, evaluation, and final grading. The prioritized features and their importance are summarized in Table 1 (Product Backlog).

Table 1. Product Backlog

ID	Name	Priority
1	Authentication	High
2	Profile	High
3	Title Submission (Pre-Proposal)	High
4	Advisor Assignment	High
5	Thesis Guidance Card	High
6	Seminar Registration	High
7	Seminar Scheduling	High
8	Seminar Assesment	High
9	Seminar Revision	High
10	Publication Article Submission	High
11	Examiner Assignment (Oral Exam)	High
12	Oral Exam Assesment	High
13	Final Thesis Grade	High
14	Submission of Graduation Completion Proof	Medium
15	Announcements	Low
16	Files	Low
17	Contact	Low
18	FAQ	Low

3.2. Sprint Planning and Test Planning

Development and testing were organized into six two-week sprints, each covering Front-End and Back-End features with QA testing. The sprint schedule was planned and tracked using Trello. Testing activities aligned with Scrum and STLC involved sprint planning to define test scope, daily executions and progress discussions, sprint reviews with bug reporting, and retrospectives to identify improvements. This iterative approach ensured continuous feedback and progressive quality enhancements throughout the development cycle.

3.3. Sprint Execution and Daily Scrum

During Sprint Execution, the team carried out development and testing activities as planned in the sprint backlog. Daily Scrum meetings were held to discuss daily progress, challenges, and plans, fostering effective team communication and quick problem resolution. This routine ensured alignment among team members and kept the sprint on track. Testing was performed in parallel with development, allowing for early bug detection and timely fixes within each sprint iteration.

Functional Testing

During sprint execution, functional testing focused exclusively on the Seminar Proposal Registration (Create) feature. Test cases were developed to verify compliance with key business rules, such as submission within the allowed period, uniqueness of the proposed title, completeness of uploaded documents, and prevention of duplicate submissions. This targeted testing ensured that the submission process functioned correctly and aligned with user expectations and system requirements. Daily Scrum sessions supported prompt communication of progress and issues, enabling rapid debugging and iterative improvements.

a. Front-End

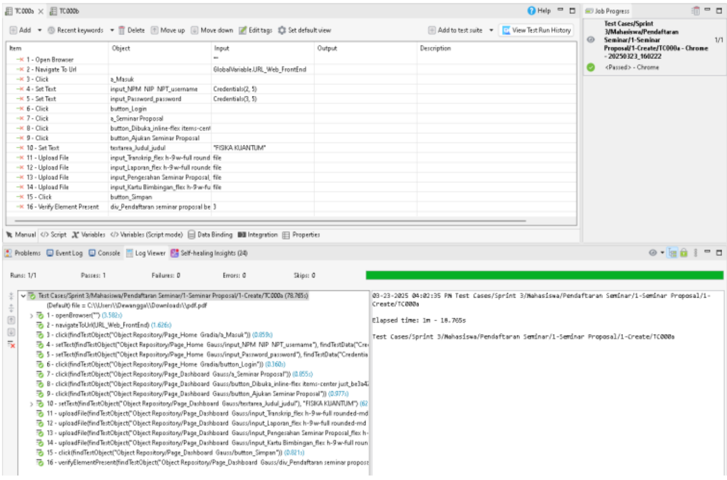

Figure 4. Positive Test Case with Katalon (Front-End)

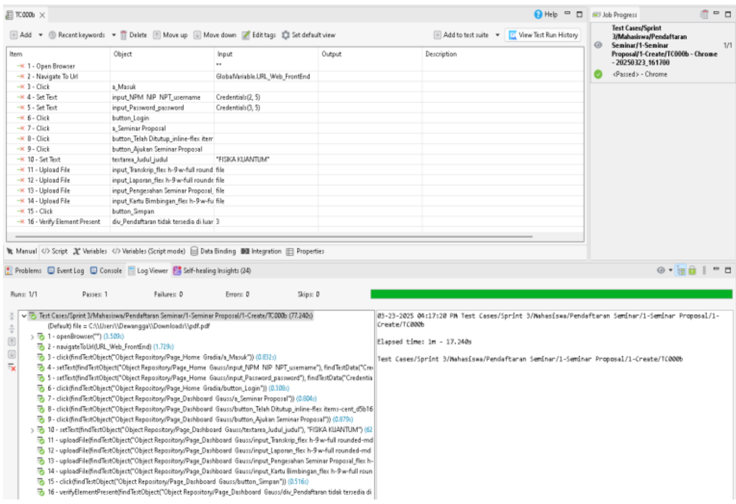

Figure 5. Negative Test Case with Katalon (Front-End)

b. Back-End

Figure 6. Positive Test Case with Postman (Back-End)

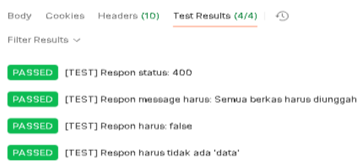

Figure 7. Negative Test Case with Postman (Back-End)

Table 2. Create Submission Seminar Proposal Test

No.	Test Scenario	Input Condition	Expected Result	Actual Result
1.	Successful submission	Valid title, within submission period, complete documents	Success	Success
2.	Submission outside allowed period	Valid input, but submission period is closed	Failed	Failed
3.	Title already used by another student	Title already exists in system	Failed	Failed
4.	Student has already submitted or submission already approved	Student previously submitted or approved	Failed	Failed
5.	Required documents are missing	One or more uploaded files are empty or missing	Failed	Failed
6.	Batch not found	Selected batch ID is invalid or not found in system	Failed	Failed

Non-Functional Testing

During the sprint execution, non-functional testing concentrated specifically on the performance aspect with a focus on scalability (1000 Virtual Users). Test cases were designed to assess the system's ability to handle varying user loads without performance degradation, ensuring smooth operation even under peak traffic conditions. This focused testing validated that the user access could scale efficiently to support increasing numbers of concurrent users, maintaining responsiveness and reliability. Continuous monitoring during development allowed early detection of bottlenecks and facilitated timely optimizations.

Figure 8. Scalability Test Script (Front-End)

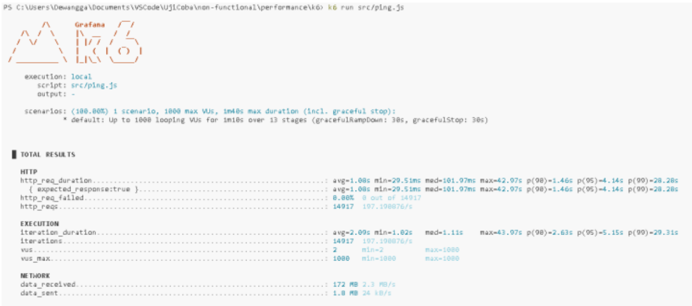

Figure 9. Scalability Test Result (Front-End)

```
export const options = {
  stages: [
    { duration: '5s', target: 10 },
    { duration: '5s', target: 50 },
    { duration: '5s', target: 100 },
    { duration: '5s', target: 200 },
    { duration: '5s', target: 500 },
    { duration: '5s', target: 300 },
    { duration: '5s', target: 600 },
    { duration: '5s', target: 1000 },
    { duration: '10s', target: 1000 },
    { duration: '5s', target: 500 },
    { duration: '5s', target: 250 },
    { duration: '5s', target: 100 },
    { duration: '5s', target: 0 },
  ],

  SummaryTrendStats: ['avg', 'min', 'med', 'max', 'p(90)', 'p(95)', 'p(99)']
};

export default function() {
  let responseBerkas = http.get('https://gradia-backend-1056453709808.asia-southeast2.run.app/api/v1/admin/berkas/');
  sleep(1);
}
```

Figure 10. Scalability Test Script (Back-End)

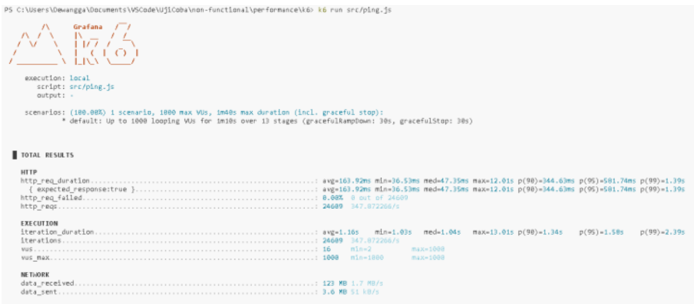

Figure 11. Scalability Test Result (Back-End)

Table 3. Non-Functional Test

No.	Non-Functional	Descriptions	Status
1.	Portability/Compatibility	The application runs smoothly on major browsers (Chrome, Safari, Firefox) and across various devices and operating systems.	PASSED
2.	Usability (Responsiveness, Readability, Accessibility)	Usability ensures the application is user-friendly by adapting layout responsively, maintaining clear readability, and meeting accessibility standards like WCAG for all users.	PASSED
3.	Performance (Scalability, Load Time, Response Time)	Performance ensures the application handles increased user (until 1000 Virtual Users) load without slowing, loads pages quickly, and responds promptly to user actions for optimal experience.	PASSED
4.	Availability	The system operates continuously without downtime to support nearly 24/7 business needs.	PASSED
5.	Security (Website Analysis, Vulnerable Testing and Penetration Testing)	The system enforces role-based access and undergoes website analysis, vulnerability testing, and penetration testing to identify and address security risks.	PASSED

Performance testing with Grafana K6 up to 1000 virtual users showed the system remained stable with a 0% request failure rate on both Back-End and Front-End. The Back-End performed responsively with low latency even at peak load, while the Front-End experienced higher response times, especially at the 95th and 99th percentiles, likely due to rendering or resource loading delays under heavy load. Overall, the system successfully handled 1000 simultaneous users without failure.

3.4. Discussion

During the testing phase, a variety of tools were used to support both functional and non-functional testing, each with its own strengths and limitations. Katalon was useful for automated functional tests but was resource-intensive. Postman simplified API testing, though it involved repetitive scripting. Grafana K6 and JMeter were used for performance testing; K6 was easier to use, while JMeter offered more flexibility through plugins but was harder to learn. Security tools like OWASP ZAP, Nikto, and Whois provided valuable insights but were less user-friendly due to command-line interfaces and high resource usage. Cross-browser tools like LambdaTest and Sauce Labs supported visual testing, with limitations depending on the account type. Lightweight tools such as Google Lighthouse, Pa11y, and Security Headers enabled quick checks for performance, accessibility, and security. Overall, the experience highlighted how hardware specifications affected tool performance and reinforced the importance of choosing tools based on testing needs and available resources.

4. Conclusion

Based on the testing conducted throughout the development of the Thesis Management System, all functional requirements for both the Front-End and Back-End were successfully verified using well-designed test cases. The testing process followed a combined Scrum and STLC approach over six two-week sprints (three months total), covering functional and non-functional needs. A total of 1,240 test cases were executed from 431 functional requirements, supporting key thesis processes such as proposal submission, supervision, seminar scheduling, evaluations, final grading, and graduation requirements. Additionally, 27 non-functional test cases were executed to assess performance, usability, availability, compatibility, and security. Various tools were used, and testing efficiency was significantly influenced by device specifications—resource-intensive tools often caused lag or crashes, highlighting the importance of proper hardware support for reliable testing. Therefore, it is recommended that future testing activities be conducted using devices with sufficient hardware specifications to ensure stable and efficient operation of testing tools. Moreover, further integration between the Thesis Management System and the university's Academic Information System (SIAMIK) is advised to achieve a more centralized, synchronized, and institutionally aligned thesis administration process.

Acknowledgement

The author sincerely thanks God Almighty for His mercy and grace, which made the successful completion of this research possible. This achievement would not have been attainable without the support and encouragement from many individuals, including family, friends, and numerous others whose contributions and assistance, though not individually named, were invaluable throughout the research journey.

References

[1] N. S. Lubis and M. I. P. Nasution, “PERKEMBANGAN TEKNOLOGI INFORMASI DAN DAMPAKNYA PADA MASYARAKAT,” Kohesi J. Sains Dan Teknol., vol. 1, no. 12, Art. no. 12, Dec. 2023, doi: 10.3785/kohesi.v1i12.1311.

[2] A. Q. Nada, Y. Fitria, and D. Darmansyah, “Pemanfaatan Search Engine sebagai Pilar Teknologi dalam Penulisan Karya Ilmiah,” J. Pendidik. Tambusai, vol. 7, no. 3, pp. 31288–31295, 2023.

[3] F. A. Nurcahyo and T. D. Valentina, “Menyusun skripsi di masa pandemi? Studi kualitatif kesejahteraan psikologis mahasiswa,” Semin. Nas. Psikol. Dan Ilmu Hum. SENAPIH, vol. 1, no. 1, pp. 136–144, Jun. 2020.

[4] R. S. Sasmita, “PEMANFAATAN INTERNET SEBAGAI SUMBER BELAJAR,” J. Pendidik. Dan Konseling JPDK, vol. 2, no. 1, pp. 99–103, Apr. 2020, doi: 10.31004/jpdk.v2i1.603.

[5] I. A. Thohir, Adittyanto, F. A. Putra, F. N. Alim, M. T. Suseno, and S. S. P. Yudha, “Dampak Perkembangan Teknologi Internet dalam Pembelajaran Jarak Jauh bagi Siswa pada Masa Pandemi Covid-19,” J. Educ. Technol., vol. 1, no. 2, Art. no. 2, Dec. 2021.

[7] T. P. Handayani, R. Maku, and R. Musa, “SISTEM INFORMASI MANAJEMEN SKRIPSI BERBASIS WEB DI UNIVERSITAS MUHAMMADIYAH GORONTALO,” J. Ilmu Komput. JUIK, vol. 1, no. 2, Oct. 2021, doi: 10.31314/juik.v1i2.1174.

[8] N. A. Ramdhan and D. A. Nufriana, “Rancang Bangun Dan Implementasi Sistem Informasi Skripsi Oline Berbasis WEB,” J. Ilm. Intech Inf. Technol. J. UMUS, vol. 1, no. 2, pp. 1–12, Nov. 2019, doi: https://doi.org/10.46772/intech.v1i02.75.
[9] S. Lena, “PERANCANGAN DAN IMPLEMENTASI SISTEM MANAJEMEN PROPOSAL SKRIPSI (STUDI KASUS JURUSAN MANAJEMEN INFORMATIKA POLITEKNIK NEGERI SAMBAS),” KLIK - Kumpul. J. ILMU Komput., vol. 8, no. 1, Art. no. 1, Feb. 2021, doi: 10.20527/klik.v8i1.373.
[10] F. S. Suwita, “Pengembangan Sistem Informasi Tugas Akhir dan Skripsi (SIMITA) di Universitas Komputer Indonesia (UNIKOM),” J. Teknol. Dan Inf., vol. 10, no. 1, pp. 71–82, Mar. 2020, doi: 10.34010/jati.v10i1.2861.
[11] S. Sintaro, “Permodelan Sistem Informasi Pembelian dan Penjualan Berbasis Website,” J. Ilm. Inform. Dan Ilmu Komput. JIMA-Ilk., vol. 1, no. 1, Art. no. 1, Mar. 2022, doi: 10.58602/jima-ilkom.v1i1.5.
[12] S. M. Prasetiyo, M. I. P. Nugroho, R. L. Putri, and O. Fauzi, “Pembahasan Mengenai Front-End Web Developer dalam Ruang Lingkup Web Development,” BULLET J. Multidisiplin Ilmu, vol. 1, no. 06, Art. no. 06, Dec. 2022.
[13] D. Susilo and P. Setiaji, “DESAIN SISTEM BACK-END PADA WEBSITE PROPERTI,” J. Tekinkom Tek. Inf. Dan Komput., vol. 6, no. 1, Art. no. 1, Jun. 2023, doi: 10.37600/tekinkom.v6i1.844.
[14] Y. Sriyeni, M. I. Mi’raj, and M. Veronica, “Evaluasi Kualitas Aplikasi Smartkit Menggunakan Metode Usability Testing,” Semin. Nas. CORIS 2022, Sep. 2022, Accessed: Nov. 21, 2024. [Online]. Available: https://corisindo.stikom-bali.ac.id/penelitian/index.php/semnas/article/view/56
[15] N. A. Sugiharto, “PENGARUH ADVERTISING DAN KUALITAS APLIKASI TERHADAP KEPUASAN PELANGGAN (STUDI KASUS PADA APLIKASI POSPAY DI KOTA TASIKMALAYA),” Mark, vol. 14, no. 1, pp. 69–76, Mar. 2024.
[16] S. S. Rahayu, W. Wahyudin, D. A. Firmansyah, and S. Susanti, “Analisis Penggunaan Tools Automation Testing pada Aplikasi : Systematic Literature Review,” REMIK Ris. Dan E-J. Manaj. Inform. Komput., vol. 8, no. 1, Art. no. 1, Jan. 2024, doi: 10.33395/remik.v8i1.13241.
[17] A. Jailani and M. A. Yaqin, “Pengujian Aplikasi Sistem Informasi Akademik menggunakan Metode Blackbox dengan Teknik Boundary Value Analysis,” J. Autom. Comput. Inf. Syst., vol. 4, no. 2, pp. 60–66, Jul. 2024, doi: 10.47134/jacis.v4i2.78.
[18] Ruliansyah, “Sekilas Tentang Software Testing Life Cycle (STLC),” Medium. Accessed: Dec. 02, 2024. [Online]. Available: https://medium.com/@ruliansyah92/sekilas-tentang-software-testing-life-cycle-stlc-1dfcd2b1f447
[19] “Complete Guide To Software Testing Life Cycle (STLC),” LambdaTest. Accessed: Dec. 02, 2024. [Online]. Available: https://www.lambdatest.com/blog/software-testing-life-cycle/
[20] R. P. Octavially, R. R. Riskiana, K. A. Laksitowening, D. S. Kusumo, M. Adrian, and N. Selviandro, “Test Case Analysis with Keyword-Driven Testing Approach on Angkasa Website Using Katalon Studio Tools,” Ultim. J. Tek. Inform., vol. 13, no. 2, pp. 134–141, 2021, doi: 10.31937/ti.v13i2.2391
[21] S. Y. Shalsabilla, E. D. Wahyuni, and N. C. Wibowo, “IMPLEMENTASI BLACKBOX AUTOMATION TESTING PADA APLIKASI DONOR MENGGUNAKAN FRAMEWORK STLC DALAM LINGKUP PENGEMBANGAN AGILE SCRUM,” JATI J. Mhs. Tek. Inform., vol. 8, no. 2, Art. no. 2, Apr. 2024, doi: 10.36040/jati.v8i2.9465.
[22] L. Mutawalli, B. K. Fathoni, and H. Asyari, “IMPLEMENTASI SCRUM DALAM PENGEMBANGAN SISTEM INFORMASI JASA DESAIN GRAFIS,” J. Manaj. Inform. Dan Sist. Inf., vol. 3, no. 2, pp. 116–122, Jul. 2020, doi: 10.36595/misi.v3i2.166.
[23] R. Gutama, “Implementasi Scrum Pada Manajemen Proyek Pengembangan Aplikasi Sistem Monitoring dan Evaluasi Pembangunan (SMEP),” AUTOMATA, vol. 2, no. 1, Art. no. 1, Jan. 2021, Accessed: Dec. 03, 2024. [Online]. Available: https://journal.uii.ac.id/AUTOMATA/article/view/17420
[24] Y. F. Kurnia and D. T. Yulianti, “Analisis Proses Software Testing pada PT. Signify Commercial Indonesia,” J. STRATEGI - J. Maranatha, vol. 3, no. 1, Art. no. 1, Apr. 2021.
[25] “Software Testing Life Cycle (STLC),” GeeksforGeeks. Accessed: Dec. 02, 2024. [Online]. Available: https://www.geeksforgeeks.org/software-testing-life-cycle-stlc/