Testing of a Classroom Facility Maintenance Monitoring System Using Alpha, Beta, and Stress Testing Methods

1st Afif Ibadurrahman Jalaluddin
Fakultas Teknik Elektro
Universitas Telkom
Bandung, Indonesia
Afifibadurrahman@student.telkomuniversity.ac.id

2nd Meta Kallista
Fakultas Teknik Elektro
Universitas Telkom
Bandung, Indonesia
Metakallista@telkomuniversity.ac.id

3rd Ashri Dinimaharawati
Fakultas Teknik Elektro
Universitas Telkom
Bandung, Indonesia
Ashridinimaharawati@telkomuniversity.ac.id

***Abstract*— Classroom facility maintenance at the TULT Building of Telkom University was previously managed manually through paper-based forms and control sheets, often causing delays, errors, and data loss. To overcome these inefficiencies, a web-based monitoring system was developed and evaluated using alpha, beta, and stress testing methods. Alpha testing validated all core features for three key user roles — admin (12 features), technician (10 features), and user (8 features) — with all functionality checks achieving a 100% success rate. Beta testing involved 30 respondents (2 admins, 2 technicians, and 26 users) who assessed usability through a 5-point Likert scale, resulting in an average score of 4.51 or 83.4%, categorized as "Very Good." Stress testing simulated 300 virtual users accessing the system concurrently, yielding a stable average response time of 224 ms, a throughput of 46 requests per second, and zero recorded errors. These evaluations confirmed that the system is functionally reliable, user-friendly, and robust under high-load conditions. The testing outcomes suggest that the system is ready for operational deployment, providing significant improvements in monitoring efficiency, real-time reporting, maintenance coordination, and data accuracy. Future enhancements are recommended, such as implementing caching mechanisms, introducing load-based authentication fallback strategies, and expanding beta testing with a more diverse and larger user group to ensure system scalability and continuous improvement.**

***Keywords*— monitoring system, alpha testing, beta testing, stress testing, web application, Telkom University**

I. INTRODUCTION

The maintenance of educational facilities in higher education institutions plays a crucial role in ensuring smooth academic operations. At the TULT Building of Telkom University, the maintenance process previously relied heavily on manual documentation and reporting, using paper forms, control cards, and verbal coordination. This traditional method was often inefficient, leading to inaccurate data records, delayed maintenance handling, and lack of historical tracking.

To address these challenges, a web and mobile-based monitoring system was developed. The system enables real-time reporting through QR code scanning, automatic technician scheduling, and centralized status tracking. This paper presents the evaluation of the system using a structured testing approach to validate its functionality, usability, and performance [1].

Three testing approaches were employed: alpha testing to verify feature implementation and error detection, beta testing to assess the user experience and system acceptance, and stress testing to evaluate the system's response under peak load. These methods aim to provide a comprehensive overview of the system's readiness for deployment and its ability to scale with demand.

II. THEORETICAL BACKGROUND

In software engineering, evaluating the quality and reliability of an information system involves various types of testing methodologies [2]. Each type of testing is designed to uncover specific attributes of the system, such as functional correctness, usability, and scalability under pressure. This section outlines the theoretical basis of three commonly applied testing techniques alpha testing, beta testing, and stress testing which were selected based on the objectives of this research. Their combined application allows for a comprehensive validation of both technical performance and user experience.

A. Alpha Testing

Alpha testing is the initial phase of functional testing conducted by the developers before public release [3]. It aims to ensure that all components of the system work according to the design. This form of testing is crucial in identifying bugs and inconsistencies within the internal environment.

B. Beta Testing

Beta testing is conducted with a selected group of users who interact with the system in realistic environments. It evaluates the usability, functionality, and user satisfaction of the system [4]. Feedback collected through Likert-scale questionnaires helps developers understand the system's strengths and areas for improvement.

C. Stress Testing

Stress testing determines the behavior of a system when it is subjected to a workload that exceeds its normal operational capacity [5]. It is an essential form of performance testing that helps identify bottlenecks, response time degradation, and failure thresholds. Tools like k6 and Grafana are often used to conduct this type of testing.

III. METHODS

Following the explanation of theoretical foundations, this section presents the step-by-step methodology implemented to test the system. The methods were applied systematically to ensure objective measurement of the system's reliability, user-friendliness, and load-handling capacity. Each testing method is detailed according to its role in validating specific aspects of the system.

A. Alpha Testing

Black-box testing was used for alpha testing across three user roles: admin, technician, and user. Each role was assigned a set of features to test:

- Admin: 12 features (e.g., dashboard access, technician management, reporting view, approval flow)
- Technician: 10 features (e.g., task acceptance, status updates, QR code processing)
- User: 8 features (e.g., complaint form submission, QR scan, ticket tracking)

All test scenarios were executed manually and validated through observation and output verification [6]. Each function was marked successful if the expected output was achieved without system error. The results showed 100% success across all test cases.

B. Beta Testing

Beta testing was conducted to collect feedback on system usability [7]. A total of 30 respondents participated, consisting of 2 admins, 2 technicians, and 26 users. Each respondent used the system to perform predefined tasks. Afterward, they filled out a usability questionnaire using a 5-point Likert scale, where 1 means "Strongly Disagree" and 5 means "Strongly Agree".

TABLE 1
Likert Scale

Value	Description
1	Strongly disagree
2	Disagree
3	Neutral
4	Agree
5	Strongly agree

The questionnaire included the following items shown in Table 3.1 Beta Testing Questionnaire Items.

TABLE 2
Beta Testing Questionnaire Items

No	Question
1	Is the feature easy to use?
2	Is the interface comfortable and easy to read?
3	Is the data entry process fast and efficient?
4	Does the system run without issues?
5	Are you satisfied with the system overall?

Average Score Calculation Formula per Question

Average Score = (Σ Score from all respondents)/(Number of respondents)

Score Interpretation
4.01 – 5.00: Very Good
3.01 – 4.00: Good
2.01 – 3.00: Adequate
1.01 – 2.00: Poor
0.00 – 1.00: Very Poor

C. Stress Testing

Stress testing evaluated the system's ability to handle a high number of concurrent users [8]. A load simulation was performed using k6, generating 300 virtual users over a 4-minute 30-second duration. The endpoint tested was responsible for fetching room master data. During the test, performance metrics including response time, request throughput, and error rate were monitored via Grafana dashboard [9].

TABLE 3
Stress Testing Configuration

Parameter	Values
Tool	K6 + Grafana
Number of Virtual Users	300 Maximum virtual users
Total Duration	5 minutes (4 minutes 30 second test + 30 second cooldown/graceful stop)
Endpoint under test	/api/monitoring/v1/mobile/complaint

IV. RESULTS AND DISCUSSION

After applying the testing procedures described previously, the results obtained are presented and analyzed in this section. Each test outcome provides insight into how well the system meets its functional and non-functional requirements [10]. The discussion also includes user responses and observed system behavior under simulated conditions.

A. Alpha Testing Results

All 30 test scenarios (12 for admin, 10 for technician, 8 for user) were executed successfully. No crashes, incorrect outputs, or user flow issues were found.

TABLE 4
Alpha Testing on Admin Role

No	Test Case	Expected Result	Actual Result	Status
1	Admin Login	System redirects to dashboard	Dashboard successfully displayed	Valid
2	Forgot Password	OTP sent to email	OTP received, password successfully reset	Valid

No	Test Case	Expected Result	Actual Result	Status
3	Dashboard	displays timeline information	Timeline displays according to data	Valid
4	Approval Menu	Displays list and allows approval/decline	Approve and decline are working, including reason input	Valid
5	Genetic Algorithm	Optimal schedule determined	Schedule successfully compiled	Valid
6	Maintenance Menu	Import schedule, verify technician work	All features work as expected	Valid
7	Master Data	CRUD tools, classes, locations, users, roles, and specialist technician	All CRUD operations successful	Valid
8	Notification	Displays all system notifications	Notifications appear in the correct order	Valid
9	Profile	Admin data edit successful	Data successfully changed	Valid
10	Logout	Exit the system and return to login	Logout successful	Valid
11	Login	The system	he system	Valid
	Logic Validation (White-box)	only accepts valid credentials, password encryption is executed	denies access without valid credentials, encryption is successful	
12	Genetic Algorithm Logic Validation	The system searches for an empty schedule intersection between the technician and class/laboratory schedules	The system successfully searches for an intersection between the technician and class/laboratory schedules	Valid

TABLE 5
Alpha Testing on Technician Role

No	Test Case	Expected Result	Actual Result	Status
1	Technician Login	Enters task page	Login successful	Valid
2	Forgot Password	Emails/tokens accepted	Password reset successfully	Valid
3	View Task	List of tasks displayed	Data is displayed according to the role	Valid
4	Detail Task	Dapat approve/decline task	Function runs well	Valid

No	Test Case	Expected Result	Actual Result	Status
5	Results Form	Can input tools and upload evidence	Data is stored and displayed in the system	Valid
6	Scan QR Code	Location of the room identified	QR read successfully	Valid
7	Availability	Availability data is stored	Data successfully entered	Valid
8	Logout	Exit the system	Successful logout	Valid
9	Validation of Login Logic (White-box)	The system only accepts valid credentials, password encryption is executed	System deny access without valid credentials, encryption works	Valid
10	Technician Role Validation	The system only displays the technician view	Validation successful, no user view	Valid

TABLE 6
Alpha Testing on User Roles

Yes	Test cases	Expected Results	Actual Results	Status
1	Login	The account was created by the admin and the login was successful	Successful login	Valid
2	Forgot Password	Reset email received	Password changed successfully	Valid
3	Scan QR Code	Complaint forms appear according to the room	Location detected successfully	Valid
4	Submit a Complaint	Complaint data sent	Complaints have successfully entered the system	Valid
5	Complaint History	All complaints appear	Complete data	Valid
6	Detail History	Display status & processes	All information appears clearly	Valid
7	Logout	Exit the system	Successful logout	Valid
8	Validation of User Access Rights	Users can only send complaints and see the repair process, they cannot access technician/admin features	System restricts access by role	Valid

TABLE 7
Alpha Testing Results

Role	Features Tested	Success Rate
Admin	12	100%
Technician	10	100%
User	8	100%

These results confirm that all system components performed according to design and were ready for real-world use.

B. Beta Testing Results

The questionnaire responses from 30 users showed a high degree of satisfaction across all usability dimensions.

TABLE 8
Usability Testing Average Score Results

Yes	Question	Average Score	Interpretation of the Score
1	Are the features easy to use?	4.47	Very Good
2	Is the display comfortable and easy to read?	4.60	Very Good
3	Is the charging process fast and efficient?	4.53	Very Good
4	Does the system run without interruption?	4.47	Very Good
5	Are you satisfied with the system as a whole?	4.47	Very Good

Calculation: (4.47 + 4.60 + 4.53 + 4.47 + 4.47)/5 = 4.51
Interpretation: Very Good

So, the overall average score of the usability testing is 4.51, which is in the Very Good category. Users reported that the system was intuitive, the layout was clear, and the response speed met expectations. These results indicate strong acceptance and minimal learning curve.

C. Stress Testing Results

System performance remained consistent under heavy load, with no errors encountered during the test window.

TABLE 7
Stress Testing Results

Metric	Value
Total Number of Requests	33,725 requests
Number of Iterations	33,725 iterations
Maximum Number of Users	300 Virtual Users
Average Response Time	224.11 ms
Maximum Response Time	1.18 ms
Median Response Time	145.56 ms
Error (Failed Requests)	0% (all requests are successful)
Throughput	124.62 request/s
Data Received	117 MB
Data Sent	28 MB

This demonstrates the system's readiness for deployment in environments where multiple users access resources simultaneously.

D. Future Development

Although the system performs well, improvements can still be made. Recommendations include:

- Implementing caching for frequently accessed endpoints
- Adding load-based authentication fallback mechanisms
- Integrating a real-time dashboard for administrators
- Expanding beta testing with more diverse user groups

V. CONCLUSION

The monitoring system for classroom maintenance at the TULT Building of Telkom University has been comprehensively evaluated using alpha, beta, and stress testing methods. Alpha testing confirmed full functionality across all roles. Beta testing revealed a high usability score of 83.4%, reflecting positive user reception. Stress testing validated system reliability under concurrent access loads. These results affirm the system's feasibility for real-world implementation and suggest that it is both technically and operationally sound.

Continued development and scaling, guided by user feedback and performance monitoring, are recommended to enhance system sustainability and efficiency.

REFERENCES

[1] M. Divyani, S. Taley, and B. Pathak, "Comprehensive Study of Software Testing Techniques and Strategies: A Review," *International Journal of Engineering Research & Technology (IJERT)*, 2020, [Online]. Available: www.ijert.org

[2] M. Hendayun, A. Ginanjar, and Y. Ihsan, "ANALYSIS OF APPLICATION PERFORMANCE TESTING USING LOAD TESTING AND STRESS TESTING METHODS IN API SERVICE," *JURNAL SISFOTEK GLOBAL*, vol. 13, no. 1, p. 28, Mar. 2023, doi: 10.38101/sisfotek.v13i1.2656.

[3] T. Menora, C. H. Primasari, Y. P. Wibisono, T. A. P. Sidhi, D. B. Setyohadi, and M. Cininta, "Implementasi Pengujian Alpha dan Beta Testing

pada Aplikasi Gamelan Virtual Reality," *KONSTELASI: Konvergensi Teknologi dan Sistem Informasi*, vol. 3, no. 1, 2023.

[4] J. Dwi Gotama, Y. Fernando, and D. Pasha, "PENGENALAN GEDUNG UNIVERSITAS TEKNOKRAT INDONESIA BERBASIS AUGMENTED REALITY," *Jurnal Informatika dan Rekayasa Perangkat Lunak (JATIKA)*, vol. 2, no. 1, pp. 28–38, 2021, [Online]. Available: http://jim.teknokrat.ac.id/index.php/informatika

[5] Indrianto, "PERFORMANCE TESTING ON WEB INFORMATION SYSTEM USING APACHE JMETER AND BLAZEMETER," *Jurnal Ilmiah Ilmu Terapan Universitas Jambi*, vol. 7, no. 2, pp. 138–149, Dec. 2023, doi: 10.22437/jiituj.v7i2.28440.

[6] A. Febrian and C. A. Vinahapsari, "Brand equity is mediated in influencing purchase intentions on e commerce," *TEST ENGINEERING AND MANAGEMENT*, 2020, [Online]. Available: https://www.researchgate.net/publication/340730724

[7] L. Schmitt, I. Haupenthal, and F. Bin Ahmed, "Website Design and Trust Elements: A/B Testing on a Start-up's Website," *ENTRENOVA - ENTerprise REsearch InNOVAtion*, vol. 7, no. 1, pp. 170–180, Dec. 2021, doi: 10.54820/zowh5239.

[8] G. Diprima, "Comparison of tools and metrics for non-functional testing of web pages," 2022.

[9] I. Ushakova, O. Plokha, and Y. Skorin, "Approaches to web application performance testing and real-time visualization of results," *Bulletin of Kharkov National Automobile and Highway University*, no. 96, p. 71, May 2022, doi: 10.30977/bul.2219-5548.2022.96.0.71.

[10] A. J. Abdulwareth and A. A. Al-Shargabi, "Toward a Multi-Criteria Framework for Selecting Software Testing Tools," *IEEE Access*, vol. 9, pp. 158872–158891, 2021, doi: 10.1109/ACCESS.2021.3128071.